Advances in the inverse biomechanical simulation framework for autonomous robotic needle steering.

Pedro Henrique Suruagy Perrusi

▶ **To cite this version:**

Pedro Henrique Suruagy Perrusi. Advances in the inverse biomechanical simulation framework for autonomous robotic needle steering.. Automatic Control Engineering. 2020. hal-02986039

HAL Id: hal-02986039
https://inria.hal.science/hal-02986039

Submitted on 2 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SURUAGY PERRUSI, Pedro Henrique
Promotion 2020
Master Thesis

Master Imagerie, Robotique, Ingénierie pour le Vivant
Option Automatique et Robotique

Spécialité TI Santé DTMI

Advances in the inverse biomechanical simulation framework for autonomous robotic needle steering

Rapport Master

ICube
Automatique Vision et Robotique (AVR)
300 bd Sébastien Brant 67412 Illkirch France

BAKSIC, Paul
p.baksic@unistra.fr

COURTECUISSE, Hadrien
hcourtecuisse@unistra.fr
+33 6 89 66 32 83

25/02/2020 - 28/08/2020

Abstract

Precise needle positioning allows interventional radiologists to perform complex therapies with reduced patient trauma. Some of the greatest challenges reported by these professionals are related to compensating for respiratory movement during needle insertion. State of the art methods use Finite Element simulations to provide robotic assistance to the needle insertion while compensating for tissue deformations. The inverse simulation is capable of steering the needle autonomously by solving an inverse problem linking the robot motion to objective functions evaluated in a biomechanical simulation. A major contribution of this study is to enhance the inverse simulation with a predictive model of respiratory motion. In addition, an asymmetric beveled tip needle model is included to add a degree of freedom to the needle tip in the inverse simulation. The results of realistic biomechanical simulations show that predictive inverse simulation can reduce both the mean value of the final position error and the amplitude of oscillations due to respiratory motion. Although the beveled tip needle model did not improve the procedure precision by itself, it can be used in future studies to enhance the control framework.

Résumé

Avancées du framework de simulation biomécanique inverse pour le pilotage automatique d'aiguilles robotisé

La radiologie interventionnelle utilise le positionnement précis des aiguilles pour effectuer des thérapies complexes en réduisant le traumatisme pour le patient. Certains des plus grands défis signalés par ces professionnels sont liés à la compensation du mouvement respiratoire pendant l'insertion de l'aiguille. L'état de l'art actuel propose l'assistance robotisée au geste d'insertion d'aiguille en compensant les déformations des tissus par biais des simulations par éléments finis. La simulation inverse est capable d'insérer l'aiguille de manière autonome en résolvant un problème inverse reliant le mouvement du robot à des fonctions objectives évaluées dans une simulation biomécanique. Une contribution majeure de cette étude est d'améliorer la simulation inverse avec un modèle prédictif du mouvement respiratoire. En complément, un modèle d'aiguille asymétrique à pointe biseautée est considéré pour ajouter un degré de liberté à la pointe de l'aiguille dans la simulation inverse. Les résultats de simulations biomécaniques réalistes montrent que la simulation inverse prédictive peut réduire à la fois la valeur moyenne de l'erreur de position finale et l'amplitude des oscillations dues au mouvement respiratoire. Bien que le modèle d'aiguille à pointe biseautée n'ait pas amélioré la précision de la procédure en soi, il peut être utilisé dans de futures études pour améliorer le cadre de contrôle.

Acknowledgements

There are many people that I would like to thank for the great experience of my master studies – which resulted in this master thesis. First I would like to thank my advisors Dr. Hadrien Courtecuisse and PhD(c) Paul Baksic for the outstanding support throughout this master thesis. Thank you Dr. Laurent Barbé for taking the time to read and review this manuscript. I also extend my thanks to Dr. Bernard Bayle for being part of my oral speech, for coordinating the TI-Santé studies at Télécom Physique Strasbourg and also welcoming me to Strasbourg as international relations coordinator.

Thank you very much to the ICube and MIMESIS teams for accepting me in their research facilities for this master thesis. Thank you Dr. Stéphane Cotin for allowing me into the MIMESIS team physical space and providing the material elements for conducting this research. A special thanks for my fellow interns and MIMESIS team researchers' patience while teaching me how to play cards at Clovis Vincent kitchen.

The experience of completing my studies in a foreign country is very challenging and rewarding. I would like to thank my girlfriend Luara Klein for joining me in this adventure of living in France. I'd also thank the warm welcome of my colleges at TI-Santé and hope we can keep in touch wherever we are worldwide. If we managed to have a great time during this last year while being separated in France, Germany and Canada, I guess we'll have no troubles in the future.

I thank as well the unlimited support my family and friends from Brazil are providing even though we're physically so far from one another.

Contents

List of Figures

List of Tables

List of Abbreviations

CT:	Computational Tomography
DoF:	Degree of Freedom
DoFs:	Degrees of Freedom
FEM:	Finite Element Model
FFC:	Feed forward control
FPS:	Frames per Second
IK:	Inverse Kinematics
LF:	Linear Filter
MPC:	Model Predictive Control
MR:	Magnetic Ressonance
MRI:	Magnetic Ressonance Image
PC:	Principal Component
PCA:	Principal Component Analysis
PCs:	Principal Components
SNR:	Signal to Noise Ratio
SVD:	Singular Value Decomposition
WFLC:	Weighted-frequency Fourier linear combiner

1 | Introduction

1.1 | Context

Minimally invasive surgeries (MIS) presents several advantages to the patient when compared to open surgery. During a MIS, surgeons employ a variety of techniques to perform a surgical procedure without large incisions to access the patient body. While these procedures effectively reduce the patient's trauma caused by the surgery, they usually require high technical skills from the surgeon. Most of the MIS involve the surgical guidance of the gesture using interventional medical imaging devices which provided limited visualization of the surgical gesture.

As an important part of the MIS, needle guided procedures have been employed to perform a variety of interventions with reduced patient trauma. One may cite needle biopsies, brachytherapy, cancerous tissue ablation and cryoablation as examples of needle guided procedures. The clinical need for assistance while performing needle punctures has been reported in the literature [1]. One of the most challenging aspects of needle positioning during interventional surgeries reported by the study is to compensate respiratory movements of the patient. It also reports most of the unwanted needle bending is reported in liver biopsy procedures.

To target this issue, a research team at Strasbourg proposes innovative strategies for an autonomous needle steering with robotic assistance. The project "SPERRY: SuPervisEd Robotic suRgerY" combines the expertise of ICube lab in robotics and MIMESIS research team in soft tissues biomechanical models. In their most recent work, an autonomous surgical procedure in moving tissue was validated in realistic simulations and published in a major robotics conference [2]. This method was first proposed by [3] and is validated in static phantom trials under several assumptions.

An overview of the clinical workflow is expressed in figure 1.1. The lower half of the diagram presents the standard MIS workflow. The standard procedure starts by the acquisition of a 3D image of the patient. This image will be used for the surgeon to plan the needle trajectory he will attempt to perform during the surgery. Based on this planning phase, the international imaging device – such as an ultrasound, fluoroscope or a C-arm – will be positioned in order to provide an adequate visualization of the surgical gesture. Finally, during the surgical operation, the surgeon

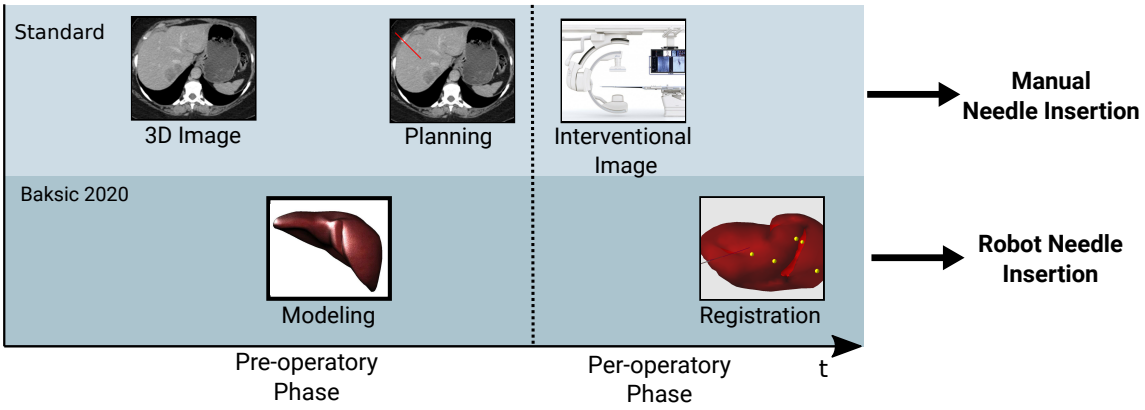

Figure 1.1: Clinical workflow proposed for minimally invasive needle insertions. The bottom view expresses the standard clinical procedure for needle biopsies and other percutaneous minimally invasive surgeries. Right above there are some required additional steps for the robotic needle insertion method proposed by Baksic et al. [2].

will perform the needle insertion while observing the medical images displayed in the screen.

The upper section of the figure 1.1 expresses additional steps which are required for the robotic needle insertion framework proposed by Baksic. Still in the pre-operatory phase, biomechanical models for the patient skin, liver and rib-cage are generated based in the medical images. In a per-operatory phase, the biomechanical models of the patient would be registered to their real position in real-time. This information is the input to drive a finite element biomechanical simulation which is as close as possible to the real surgical scene and will command the robotic needle insertion. Adagolodjo and Baksic denotes this simulation as the inverse simulation.

It is a highly specialized scope of research to propose improvements to the inverse simulation framework. Multiple solutions to the finite element models (FEM) simulation per second must be computed in order to achieve stability of the inverse simulation. Recently Baksic proposed a numerical optimization strategy to accelerate even further these simulations and was able to perform the needle insertion in moving targets.

1.2 | Objective

This study aims to investigate additional features which could improve the automatic needle insertion framework based in inverse simulations. A first and major contribution of this study is to integrate motion prediction filters to the inverse simulation control framework. As the simulation runs in a predictive state of the biomechanical models, it's expected that motion disturbances would be compen-

sated. The final needle insertion precision will be evaluated. All results are acquired in realistic biomechanical simulations.

Another features also explored by this study are the addition of controllable degrees of freedom (DoF) the needle tip and to the robotic manipulator. The addition of a beveled tip needle to the needle insertion solution is expected to increase the reactivity of the system and reduce final positioning errors. On another hand, adapting the inverse simulation solution to a redundant 7 DoF robotic manipulator model can augment its trajectory alternatives to reach the control framework objectives.

While the groundwork for each solution has been development in this master thesis, only the respiratory motion prediction filters and the beveled tip needle model have been validated and tested in the inverse simulation framework. These two contributions are described in detail in chapter 4 and their results and discussions are listed in chapter 5. The redundant robot inverse kinematics strategy has been developed but it has not yet been integrated into the inverse simulation framework for needle steering. Details on these developments and preliminary results are in appendix A.

2 | State-of-the-Art

This state-of-the-art review will address each of the objectives of this study. Section 2.1 addresses the mathematical model for needle insertion. Section 2.2 reviews respiratory motion compensation strategies for medical robots in radiotherapy and in needle steering.

2.1 | Needle Tissue Interaction Model

There are multiple approaches to modeling the interaction between needle and tissue. The geometry of the needle's tip is specially important in those models. While a symmetric needle tends to cut the tissue in a straight direction, a beveled tip needle induces is subjected to asymmetric reaction forces. As a result, a beveled tip needle tends to present a deflection in a privileged direction.

2.1.1 | Kinematic Model

The kinematic unicycle model proposed by Webster et al. [4] is a widely used model of beveled tip needle deflection. It proposes a geometrical perspective of the needle tip trajectory based in its previous states. According to them, the needle tends to follow a circular trajectory which is function of the insertion velocity v_{ins}, the needle axial rotation u and the needle path curvature κ which needs to be identified. Moreira et al. presented an online estimation of κ based in the visual tracking of the needle and a Kalman Filter [5]. This needle model is specially used for duty cycle control approaches [6], in which the desired deflection of the needle is achieved by constantly applying rotation to its shaft u. When a straight path is desired, the needle rotates in a higher frequency. When no rotation is performed, the needle deflection tends to its maximum value κ.

The kinematic model posses some significant limitations [7]. First, this deflection model is not suitable for symmetric needles, since it would always assume a straight trajectory in the tissue. Also, a strong assumption is made: the tissues are stationary and all the forces exchanged along the needle axis are directly transmitted to the needle tip. These assumptions stand only when the tissue does not induce motion

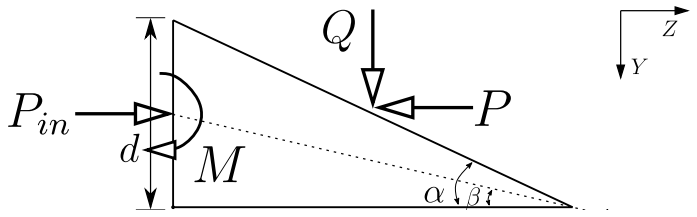

Figure 2.1: Needle tissue interaction model's inputs and outputs proposed by Misra et al. [9]. The dashed arrow represents the direction of advance of the needle in the tissue caused by the input force P_{in}.

to the needle. It is also assumed that the needle is very flexible with respect to the tissue, which limits its usage in the presence of intense physiological disturbances. In order to account for lateral motion, an extension has been proposed to this kinematic model [8].

2.1.2 | Mechanical Energy Formulation

Another needle deflection model widely used in the literature is an energy based formulation, proposed by Misra et al. [9]. First a model for the forces observed at the level of the needle tip is derived. Figure 2.1 presents a closer look into the geometry and forces involved on this model. The model takes as inputs the beveled tip needle geometry: the needle diameter (d), bevel angle (α) and two medium-dependent properties: the cut angle (β) and the needle-tissue interaction stiffness (K_T). The cut angle β depends on both the needle geometry and the medium. Misra usually combines it with the bevel angle, using their ratio as a single variable (β/α). So, given the needle diameter and the needle-tissue stiffness – which can be measured – the only parameter left to identify is the ratio β/α. In their work they explore $\frac{\beta}{\alpha}$ in an interval from 0.25 up to 1.0

As intermediate variables, various dimensions of the needle geometry are computed as follows: $a = \frac{d}{tan\alpha}$, $b = \frac{d}{sin\alpha}$, $e = d - atan\beta - \frac{b}{3}sin\alpha$ and $\theta = 90° - \alpha$. These parameters are used to compute the resultant axial force P, transverse force Q and resulting torque M according to the equations 2.1, 2.2 and 2.3. It is important to notice these forces remain constant as long as the needle is inserted into the medium.

$$P = P_{input} - \frac{K_T b^2}{2} sin(\alpha) tan(\alpha - \beta) \tag{2.1}$$

$$Q = \frac{K_T a^2}{2} tan(\beta) - \frac{K_T b^2}{2} cos(\alpha) tan(\alpha - \beta) \tag{2.2}$$

$$M = \frac{K_T b^2}{2} \left(\frac{b}{3} cos^2(\alpha) - e sin(\alpha) \right) tan(\alpha - \beta) - \frac{K_T a^3}{6} tan(\beta) \tag{2.3}$$

Misra then makes use of the calculus of variations Rayleigh-Ritz method to compute a mechanical energy formulation for the shape of the needle which minimizes the energy stored in the system. This method is suitable to model both symmetric needles and beveled tip needles.

Chevrie et al. adapted this energy formulation with virtual springs to account for lateral disturbances in moving soft tissues [10]. They've defined

$$l = \left\{ l \in \mathbb{R} \mid -L_{free} \leq l \leq L_{ins} \right\}$$

the curvilinear metric coordinate parameter along the needle path. By definition $l = 0$ at the insertion point, L_{free} is the length of the needle outside the tissue and L_{ins} is the length of the needle inside the tissue. The model is composed by two spline curves, one for the needle $\mathbf{c}^N$ and one for the tissue's cutting path resting position $\mathbf{c}^T$. The needle is modelled as a one dimensional beam represented as $\mathbf{c}^N(l) \in \mathbb{R}^3$ at the coordinate l as expressed in equation 2.4. Matrix $\mathbf{M}_i \in \mathbb{R}^{3X(r+1)}$ contains the coefficients of the polynomial curve $\mathbf{c}_i^N$ and $\chi_{\mathbf{c}_i^N}(l)$ is the characteristic function of the curve.

$$\begin{aligned} \mathbf{c}^N(l) &= \sum \mathbf{c}_i^N(l) \\ \mathbf{c}_i^N(l) &= \chi_{\mathbf{c}_i^N}(l)\ \mathbf{M}_i \begin{bmatrix} 1 & l & \dots & l^r \end{bmatrix}^T \end{aligned} \tag{2.4}$$

As the model accounts only for small deformations along the needle axis – which should be assured to avoid tissue damage – an elastic deformation law is used to describe the resulting force in each segment of the needle. This relation is expressed for a given length $l \in [l_1, l_2]$ in the needle's length by equation 2.5.

$$\mathbf{F}(l_1, l_2) = -K_T \int_{l_1}^{l_2} \mathbf{c}^N(l) - \mathbf{c}^T(l)\ dl \tag{2.5}$$

Through this relation it is possible to express the elastic energy stored in the tissue through the entire inserted length L_{ins} as in equation 2.6. Another important relation which could be derived is the bending energy of the needle according to the Euller-Bernoulli beam model as expressed in equation 2.7 where E is the Young's modulus of the needle and I is its second moment of area. According to [11] E_T et E_B are sufficient to represent the quasi-totality of the energy stored in the needle steering system.

$$E_T = K_T \int_0^{L_{ins}} \left\| \mathbf{c}^N(l) - \mathbf{c}^T(l) \right\|^2\ dl \tag{2.6}$$

$$E_B = \frac{EI}{2} \int_{-L_{free}}^{L_{ins}} \left(\frac{d^2 \mathbf{c}^N(l)}{dl^2} \right)^2\ dl \tag{2.7}$$

The resolution of the model equations is based in the Rayleigh-Ritz method for the calculus of variations. Through the minimization of the energy of the system under constraints as expressed in equation 2.8 we may compute estimate positions for each needle points along its shaft. The solution of the optimization is a vector ($\mathbf{m}$) containing all spline coefficients of matrices $\mathbf{M}_i$. Matrix $\mathbf{A}$ and vector $\mathbf{b}$ contains second order coefficient degrees, position and velocity constraints imposed by the needle holder. As a result, the model is able to converge and find new needle curve positions at each of its updates.

$$\begin{aligned} \min_{m} \quad & E_B + E_T \\ \text{s.t.} \quad & \mathbf{Am} = \mathbf{b} \end{aligned} \tag{2.8}$$

2.1.3 | Finite Elements Model

Finite Elements Model (FEM) is a powerful method to provide numerical solutions to partial differential equations and model complex mechanical systems. The method is able to reach solutions for a finite spatial-sampling mesh of the objects being modeled. As more details are included in the model, i.e. finer meshes, more computational cost is added to the solution of these models. In addition, the type of interaction modeled between the needle and the tissue can add a significant complexity to the system [12]. In a general way, these methods are hard to apply in real-time context since it depends on numerical optimizations and model reductions to reach a compatible numerical solver frequency. Recent research shows, however, promising perspectives of using fast FEM simulations to perform needle steering in soft tissues [2].

Multiple needle models were proposed to FEM simulations. Di Maio in 2003 models a needle as a single rigid 1D beam [13]. In 2005, he enhances the model with flexibility in interactive simulations [14] followed by Chentanez et al. [15]. A succession of rigid beams were proposed by Goksel [16] to model a flexible needle. In all these techniques, when needles cuts through the tissue model a remeshing step is necessary, which induces potential stability issues.

To solve this issue, Duriez et al. proposes a solution capable of modeling the needle and tissue interaction as Lagrangian multiplier constraints along the needle path [17]. A bilateral Lagrangian constraint is applied at the needle insertion point, which limits its displacement in all directions. During the needle insertion, trajectory constraints are applied along the needle path to limit its displacements in the 2D plan normal to the insertion axis. In addition to that, friction constraints may be applied in order to model the tissue reaction to the axial motion of the needle. These constraints are mapped to the tissue model through their barycentric coordinates with no need to re-meshing the tissue model. For more information in the solution of the FEM model with Lagrangian constraints, be sure to refer to [18] and section 3.1.

2.2 | Respiratory Motion Compensation

Robot assisted tracking of the respiratory motion is an active field of research in medical robotics. Medical devices makes use of motion models to track in real time internal targets in a patient's body. These have specially been used in radiotherapy treatments in order to minimize the radiation exposure of the patient. Motion compensation is also an essential feature of robot assisted procedures in the thorax or abdomen, such as laparoscopic procedures [19] [20], endoscopy [21], cardiac interventions [22] [23] and percutaneous liver interventions [24] [2].

As a study of the basic control methods more generally suited to follow respiratory movement, Arenbeck et al. proposed a comparison between five different predictive control approaches to compensate respiratory disturbances [25]. As a conclusion, the usage of feed forward control (FFC) and model predictive control (MPC) strategies is recommended to perform position-based motion compensation.

In radiation therapy, major manufacturers of medical devices have proposed solutions specifically designed to account for respiratory motion. One intuitive strategy is to activate the radiation beam only when the target tumor is expected to receive maximum radiation dosage – and leave it off in all other moments. This strategy is called respiratory gating [26], where if specified conditions are met, a binary control signal is sent to the device to perform the therapy. As commercially available examples one may cite the Respiratory Gating for Scanners [1], AZ-733V [2] and Catalyst [3]. Gating techniques present, however, important drawbacks in terms of increasing the therapy duration and their transposition outside radiation treatment is difficult [27].

Another strategy to compensate the breathing motion during surgical procedures is the real-time tracking of the motion model outputs. Systems which adopt this strategy usually makes use of robotic actuators to either update the laser emitter position [28], or move the patient instead using a robotic couch [29]. According to [27], some of the most effective solutions to the motion compensation when tracking a specific target in radiotherapy could be described as follows:

1. Detection of a surrogate signal (usually medical images)
2. Estimation of the actual target position through a correlation model
3. Prediction of position after latency interval
4. Treatment adaptation (or robot movement)

Some interesting solutions to model and compensate the breathing dynamics were presented by the CyberKnife robotic system designed for stereotactic radiosurgery

[1] Varian Medical Systems, Palo Alto, CA, USA
[2] Anzai Medical, Tokyo, Japan
[3] C-RAD AB, Uppsala, Sweden

[30] [31]. Their respiratory tracking system Synchrony uses external optical markers to reduce the radiation exposure of the patient and a more precise treatment [32] [33] [34]. Their system is able to estimate real time tumor position to perform precise radiotherapy in the brain lung and pancreas [35] [32]. CyberKnife is approved for treatment of non-stationary tumors by the United States Food and Drug Administration since 2001 and Syncrony was approved in 2004.

2.2.1 | Motion compensation in needle steering

In needle steering procedures, the surgical gesture induces movement to the organs. Along the needle insertion path, the tissue is displaced and may even be damaged by the needle. This represents a relevant difference when compared to the radiology motion models presented in section 2.2, since the radiation emission does not impose any movements to the patient. So in order to effectively compensate physiological motion in needle steering, a needle tissue interaction model must be considered in order to perform effective feedback control [36].

A method for prediction and active compensation of respiratory motion was first proposed by Riviere et al. in 2001 [37]. They intend to perform kidney punctures with robotic assistance in free-breathing patients with similar precision to breath holding techniques. Although a complete surgical workflow is proposed to enhance the surgical precision, only results of the respiratory prediction filter precision are presented. So this work actually reassembles most the radiotherapy algorithms for motion prediction presented by Johl et al. [38]. As they do not account for needle tissue interaction, further experiments are necessary, since their prediction accuracy may not reflect a final needle position improvement. Similar techniques were later used to perform motion compensation in laparoscopy and beating heart surgeries [39] [19] [40].

In 2015 researchers of the University of Twente [41] claim to be the first to propose an experimental setup for needle steering in a moving soft tissue. In their test-bed they produce sinusoidal disturbances in 1 degree of freedom (DoF) of a moving platform which supports a gelatin phantom. Disturbances are only generated in the axial direction of the needle insertion, which is a strong assumption since lateral motions induces the most significant errors [10]. Their control strategy is based in ultrasound visual servoing detailed in [42]. Tissue motion is estimated using a direct application of Hooke's law using readings from a force sensor next to the insertion point. Motion compensation is achieved using a constant velocity Kalman filter estimation of the disturbance position and its velocity.

Dealing with lateral tissue motion presents an even greater challenge to motion compensation in robotic needle steering. At University of Rennes Chevrie et al. propose a flexible needle 3D model capable of predicting the needle shape deflection caused by lateral motion disturbances [10] [43]. This model was presented previously in section 4.1.2 as an adaptation of Misra's mechanical energy formulation [9]. The only input needed by their needle model to compute a needle deflection which min-

imizes equation 2.8 is the robot odometry information. In their first work in 2015, Visual measures of the 3D position multiple points along the needle shaft are combined with the model outputs in an Unscented Kalman Filter (UKF) [44] to provide model updates in the tracked needle positions along its shaft.

In their following work, researchers of Chevrie et al. enhanced their lateral motion compensation strategy through a novel control law which accounts to minimize the lateral forces exerted into the tissue during the needle insertion [43]. Their first approach in a hybrid control law which accounts for the needle and tissue interaction model [9] was first introduced in 2016 [45]. In 2018, they've combined this control law with their motion compensation strategy [10] to accomplish a needle steering control strategy in moving tissues which minimizes tissue damage.

Researchers at the University of Strasbourg are investing in a novel inverse FEM simulation to command needle steering in soft tissues. Their method is able to use visual measures of fiducial markers attached to the liver as inputs to recreate a virtual simulation of the surgical scene – denoted inverse simulation. From this virtual scene, it's possible to derive appropriate commands for a robotic needle manipulator to steer needles along a predefined surgical path. Although the first approaches were validated in simulated and real world experiments, the method's computational complexity limited its application in scenarios where physiological motion is present [3] [12]. Only in their most recent work, Baksic et al. proposed numerical optimization and stabilization techniques to accomplish needle steering under respiratory motion disturbances [2].

Both the needle FEM model and the motion compensation literature presents interesting features which could be added to Baksic's inverse simulation control framework. This study proposes a first enhancement to the inverse simulation with an implementation of a beveled tip needle needle-tissue interaction model. An adapted version of the energy formulation proposed by Misra et al. [9] will be integrated into the needle steering solution. In addition to that, a prediction model for respiratory motion compensation based in Johl et al. [38] prediction filters will also be proposed. The technical details of this work are described in section 4.

3 | Background

This section offers a technical introduction on the laboratory solutions to perform iterative biomechanical simulations. Also, additional information related to the needle steering strategy proposed by Baksic et al. will be explored to introduce the concept of the inverse simulation.

3.1 | Biomechanical FEM Simulation

Traditionally Finite Element Model (FEM) simulation of soft body mechanics requires computational costs which exceeds real-time application requirements. This is notably the case of the mechanical engineering simulation software Altair Hyperworks [1] and COMSOL [2]. In order to use these simulations to drive robotic commands, the simulation should be able to operate with as minimal delay as possible.

In order to achieve both high-performance biomechanical simulations and realistic behavior, a linearized implicit integration scheme is used alongside with several numerical optimization strategies. An execution rate of $20-40Hz$ of the biomechanical simulations is desired for online applications. Following [18], Newton's second law applied to a rigid body can be expressed in terms of Lagrange multipliers as in equation 3.1.

$$\mathbb{M}(\mathbf{p}, \dot{\mathbf{p}}) = \mathbb{P}(t) - \mathbb{F}(\mathbf{p}, \dot{\mathbf{p}}) + \mathbb{H}(\mathbf{p})^T \lambda \tag{3.1}$$

The generalized degrees of freedom of the system's are $\mathbf{p} \in \mathbb{R}^n$. In the biomechanical simulation case they should be read as the mesh's node positions. $\mathbb{M}(\mathbf{p}) : \mathbb{R}^n \rightarrow \mathcal{M}^{n \times n}$ is the inertia matrix, $\mathbb{F}$ represents the internal forces applied to the simulated object and $\mathbb{P}$ contains the external forces. $\mathbb{H}(\mathbf{p})^T$ is a function which outputs constraint directions in the simulation depending on the object positions and λ is a vector of Lagrange multipliers containing the force intensities of these constraints.

The inertia matrix $\mathbb{M}(\mathbf{p})$ and internal forces $\mathbb{F}$ are derived from the biomechanical simulation framework SOFA. Components denoted *ForceFields* are responsible to

[1] https://altairhyperworks.com/

[2] https://www.comsol.fr/

model these interactions with the rigid or deformable model for each of the simulated objects. The co-rotational formulation was used as deformation model in all our simulations, since it offers a good trade off between numerical accuracy and computation effort.

The solution of a FEM simulation with Lagrangian constraints is not trivial. Courtecuisse et al. proposed graphical processing unit (GPU) optimizations to retrieve a fast convergence for the implicit integration scheme with Lagrangian constraints [18]. The interaction between two non-rigid bodies requires solving the Karush-Kuhn-Tucker (KKT) system [46] expressed in equation 3.2. For each body $n \in \{1, 2\}$, needle and tissue, respectively, $\mathbf{A}_n$ is defined as the stiffness matrix and contain the linear elastic properties of the model, alongside with it's innercy and damping. $\mathbf{b}_n$ stands for the internal and external forces exerted to each model and $\Delta\mathbf{v}_n$ is the variation of velocities during a time step. $\mathbf{H}_n$ stands for the *Jacobian of Constraints* and encode the constraint directions in the Cartesian motion space. At the end of the implicit integration time step the model forces λ which minimizes the constraint violations δ are computed interactively.

$$\begin{cases} \mathbf{A}_1\Delta\mathbf{v}_1 + \mathbf{H}_1\lambda = \mathbf{b}_1 \\ \mathbf{A}_2\Delta\mathbf{v}_2 + \mathbf{H}_2\lambda = \mathbf{b}_2 \\ \mathbf{H}_1\mathbf{v}_1 + \mathbf{H}_2\mathbf{v}_2 = \delta \end{cases} \tag{3.2}$$

The matrices $\mathbf{H}_n$ play a major role in this FEM simulation framework, since the interaction between objects, registration constraints and even robotic objective functions are modeled as Lagrangian constraints. The needle tissue interaction model used to model the contact between the two bodies n is encoded as a bilateral constraint at the insertion point and trajectory constraints along the inserted needle's path [17].

3.2 | Inverse Simulation

As introduced by Adagolodjo [12], the needle insertion is interpreted as a minimization problem. A cost function composed of several objective functions should be minimized to guide the needle along a predefined path – explained in depth in section 3.3. In order for these objective functions to account for the non-linear deformations imposed by the robot to the patient's body, a real-time biomechanical simulation needs to run in parallel to the real robotic insertion. This is what Adagolodjo and Baksic refer to as the inverse simulation.

The inverse simulation should replicate as closely as possible the real surgical scenario, while efficient enough to run with $20to40Hz$ frequency. As the clinical overflow of figure 1.1 describes, the biomechanical models and planning acquired in the pre-operatory phase are used during the per-operatory phase to drive the inverse simulation. It takes as inputs the current robot position and the position of fiducial markers attached to the robot base, the needle base and the patient's liver to

register itself with respect to the physical world. So it is assumed the operating room disposes of a 3D localization system capable of detecting the fiducial markers attached to the robot base, the needle base and also an interventional imaging solution capable of retrieving the 3D position of the fiducial markers attached the patient liver. This method has not yet been validated under a complete setup such as the described, only in laboratory conditions with a static phantom, where the fiducial marker localization and registration are assumed to be solved problems.

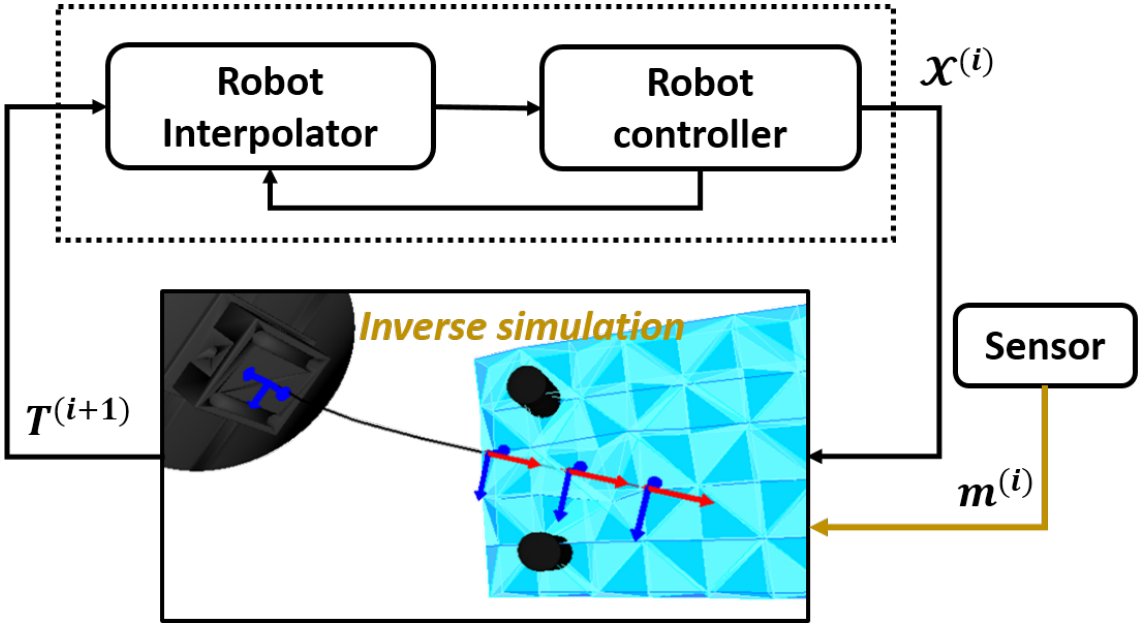

Figure 3.1: Adapted from [2], expresses the inverse simulation as the feedback loop to derive robotic commands for the needle steering. The robot interpolator and robot controller are assumed to be embedded into the robot's controller architecture.

In order to derive robotic commands to follow moving organs on the patient's body, the inverse simulation should be as fast and precise as possible. Baksic's latest contribution to this control strategy was an optimization of the inverse simulation loop. As expressed in figure 3.1, the inverse simulation loop is responsible for closing the robotic command loop. A faster loop implies more reactivity to the control strategy, allowing it to remain stable with moving targets, when its predecessors diverged. In order to keep the biomechanical simulation simple, the only model used is the patient's liver, as expressed in figure 3.2. Information about other organs, such as the skin entry point need to computed externally, by a 3D localization system, for example.

3.3 | Control Framework

As control strategy Adagolodjo and Baksic use a task optimization strategy [47] to linearize around the current state of the FEM and estimate the Jacobian of the sim-

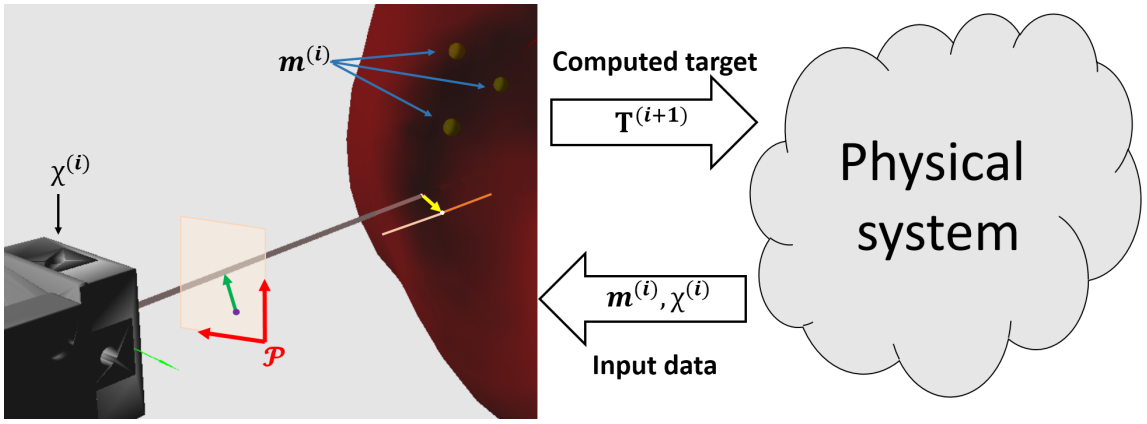

Figure 3.2: Figure adapted from [2], represents the inverse simulation, it's physical world inputs and outputs. At the arbitrary time step i, it receives as inputs the current robotic end effector position $\chi^{(i)}$ and a set of fiducial marker positions $\mathbf{m}^{(i)}$. Also, the needle insertion position $\mathcal{P}$ into the patient's body is assumed to be known by the inverse simulation. The green arrow represents the constraint objective function $\mathbf{e}_e$ and the yellow arrow stands for $\mathbf{e}_p$. As output, the inverse simulation expresses the next robot end effector position.

ulation $\mathbf{J}_6$ which minimizes a set of objective functions[3]. A key aspect of Baksic's contribution is to model these objective functions as virtual Lagrangian constraints, forming *constraint objective functions*. We'll denote the constraint objective functions as $E(\mathbf{q}, \chi, \mathbf{m})$, a function of the robot joint position $\mathbf{q}$, the robot end effector Cartesian position χ and the measured fiducial marker position $\mathbf{m}$. Values of this objective function for a specific set of input variables will be expressed as $\mathbf{e}$.

Based on this new approach, they are able to take advantage of the FEM interaction solver not only to account for the collision model between multiple bodies in the scene, but also find the optimal Cartesian position of the robot which minimizes $E(\mathbf{q}, \chi, \mathbf{m})$. To avoid introducing errors to solution for the mechanical constraints, the forces associated to each constraint objective functions are artificially forced to 0. The following expressions are used as constraint functions in their control framework:

1. $\mathbf{e}_p \in \mathbb{R}^3$: **needle position along the predefined path** minimizes the error between the needle current position $\mathbf{q}_{tip}$ and its trajectory points $\mathbf{q}_{target}$.

$$\mathbf{e}_p = \mathbf{q}_{target} - \mathbf{q}_{tip} \tag{3.3}$$

2. $e_a \in \mathbb{R}$: **needle orientation outside of the patient**. Is measured between the normalized directions of the needle tip $\mathbf{n}_{tip}$ and the tangent to the trajectory at the entry point of the skin.

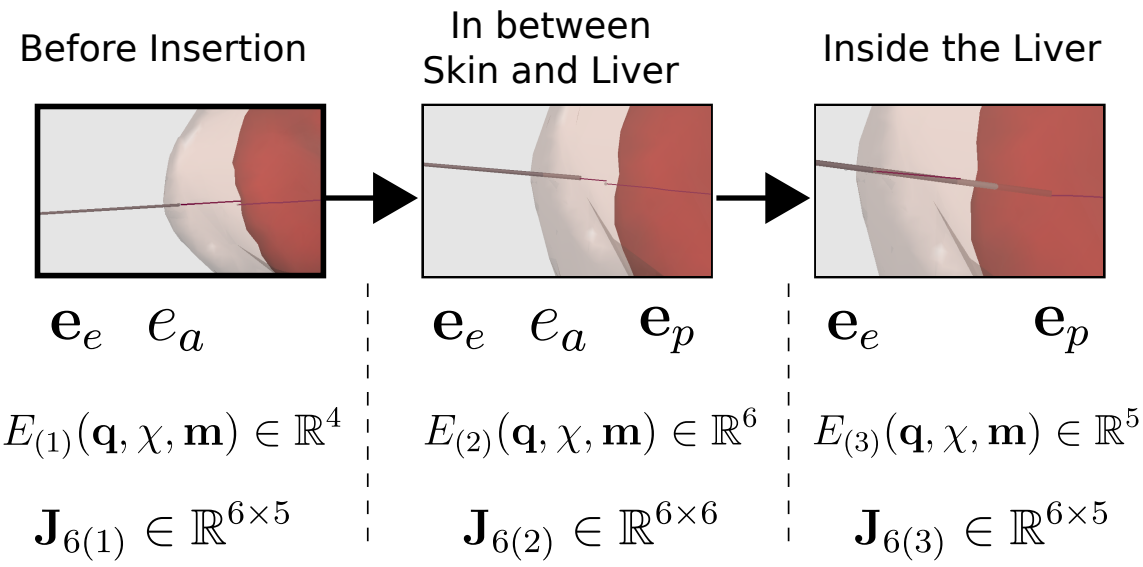

Figure 3.3: Dimensions of objective functions $E(\mathbf{q}, \chi, \mathbf{m})_{(n)}$ and Jacobian of the simulation $\mathbf{J}_{6(n)}$ for each of the $n \in [1, 2, 3]$ insertion phases.

$$\mathbf{e}_a = arccos(\mathbf{n}_{tip} \cdot \mathbf{n}_{traj}) \tag{3.4}$$

3. $\mathbf{e}_e \in \mathbb{R}^2$ **motion of the entry point** should be minimized. It is defined as a two dimensional vector of the distance between the actual penetration point $\mathbf{p}_c$ and its position stored at the moment of the puncture $\mathbf{p}_p$ projected in the surface of the skin $\mathcal{P}$.

$$\mathbf{e}_e = proj_{\mathcal{P}}(\mathbf{p}_c - \mathbf{p}_p) \tag{3.5}$$

Not every objective constraint function is active during the entire needle insertion phase. Before the needle insertion only $\mathbf{e}_p$ and e_a are active, to guide the robot alongside the trajectory with the appropriate insertion angle. So in this first phase the Jacobian matrix possesses dimensions $\mathbf{J}_{6(1)} \in \mathbb{R}^{6\times4}$. When the needle is inserted in the patient skin but has not yet reached the patient's liver, all the objective constraint functions are active and $\mathbf{J}_{6(2)} \in \mathbb{R}^{6\times6}$. Finally, when the needle is inserted through skin and also through the liver, the needle orientation on the outside e_a constraint is dropped, leaving $\mathbf{J}_{6(3)} \in \mathbb{R}^{6\times5}$. This behavior is summarized in figure 3.3.

A Jacobian matrix $\mathbf{J}_6$ between the robot end effector position and the objective functions still needs to be computed in order to derive commands from their solutions. A strategy to compute it is to apply virtual perturbations $(\Delta\chi^j)$ of the robot joint positions in the simulation. After each perturbation, a new solution of the objective constraints E is computed for this new configuration. The Jacobian matrix can be estimated using the finite differences of E over $\Delta\chi^j$, as expressed in equation

3.6. This formulation requires 12 simulation steps to account for each positive and negative disturbance of $\Delta\chi^j$ all 6 DoF of the robot.

$$\mathbf{J}_6^j = \frac{E(\mathbf{q}, \chi + \Delta\chi^j, \mathbf{m}) - E(\bar{\mathbf{q}}, \bar{\chi} - \Delta\chi^j, \mathbf{m})}{2||\Delta\chi^j||} \tag{3.6}$$

The inversion of the Jacobian matrix is performed alongside with a Tikhonov regularization [48], in order to prevent the ill conditioning of the matrix. This operation is performed via singular value decomposition of $\mathbf{J}_6^T\mathbf{J}_6$ to obtain the matrices $\mathbf{UDV}^T$. So the expression of the Jacobian's pseudoinverse is given in equation 3.7. The matrix $\mathbf{\Delta}_{\sigma(\mathbf{J}_6^T\mathbf{J}_6)<r}$ is a diagonal matrix with each diagonal element $\Delta_{k,k}$ is function of a diagonal element of matrix $\mathbf{D}$ $d_{k,k}$ and a regularization parameter r, as expressed in equation 3.8.

$$\mathbf{J}_{6r}^+ = (\mathbf{J}_6^T\mathbf{J}_6 + r\mathbf{U}\mathbf{\Delta}_{\sigma(\mathbf{J}_6^T\mathbf{J}_6)<r}\mathbf{V}^T)\mathbf{J}_6^T \tag{3.7}$$

$$\Delta_{k,k} = \begin{cases} d_{k,k}, \text{ if } d_{k,k} > r \\ r \text{ otherwise} \end{cases} \tag{3.8}$$

Finally, operational-space end-effector target positions $\mathcal{T}^{(i+1)}$ can be computed based in the previous Jacobian, objective function estimates and a control gain $\mathbf{k}$. The control gain is applied over the objective functions using a Hadamard product $\odot$.

$$\mathcal{T}^{(i+1)} = \chi^{(i)} - \mathbf{J}_r^+(\mathbf{k} \odot \mathbf{e}^{(i)}) \tag{3.9}$$

4 | Methodology

This chapter presents each of the contributions of this master thesis. It is organized as a section for each different contribution. First the respiratory motion prediction filters and their integration to the inverse simulation framework will be described. Then the beveled tip needle model implementation and its SOFA implementation is detailed.

4.1 | Respiratory motion prediction filters

Based on the extensive research of Ernst and Johl's into respiratory prediction filters for robot-assisted radiotherapy, these strategies can potentially be transposed to the robot-assisted needle steering field. By adding a predictive component to the fiducial marker tracking in the inverse simulation, Baksic's control framework could partially compensate the delays caused by slow sensors and the FEM simulation computational time. If the inputs of the inverse simulation can be correctly estimated a time-horizon before, the robustness of the motion compensation aspect of Baksic solution can be improved even further.

As discussed in section 2.2, Johl et al. presented in their survey a comparison between 18 prediction filters applied to extensive CyberKnyfe clinical data [38]. First, they perform a dimensionality reduction over the principal components (PCs) of the marker 3D positions [49] [50]. Then the prediction performance is evaluated between the predicted PC and the reference signal.

4.1.1 | Linear Prediction Filter

Given the excellent results reported by Johl et al. and its intuitive nature, the linear filter was the chosen solution to perform this study. A first step to integrate signal prediction to the inverse simulation framework is to compute the PC of each fiducial marker position – as expressed in the diagram of figure 4.1. So, as the patient breathes, each marker 3D position $\mathbf{P(t)}$ is stored in a rolling buffer $\mathbf{X} \in \mathbb{R}^{3xWP}$. In this preparation phase some cycles of the patient respiration are observed right before the needle insertion procedure. As soon as the buffer is filled, a singular value decomposition (SVD) is performed over the centralized marker positions.

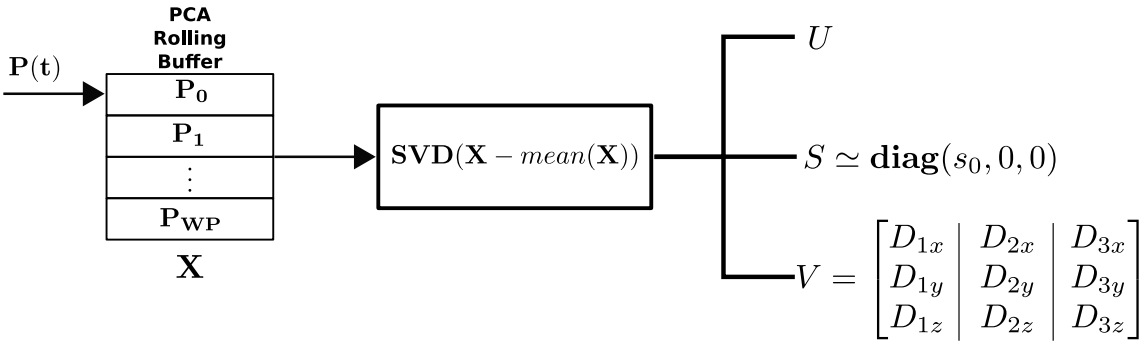

Figure 4.1: Preparation phase for the linear prediction filter. A rolling buffer of size WP stores the Cartesian position of each marker on the previous time steps P_i, $i \in [0, WP]$. An singular value decomposition (SVD) is applied to the centered rolling buffer and the first singular value usually overweighs the remaining values [49]. The principal components of the signal are the columns of the V matrix [51].

$$\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \tag{4.1}$$

It is expected that the first singular value presents itself more significantly than others $\Sigma \simeq diag(s_0, 0, 0)$ [49]. This assumption has only been validated for external fiducial markers, though we expect to verify a similar behavior. As a result, each principal direction are given by the columns of the orthogonal matrix $V = \begin{bmatrix}\mathbf{D_1} & \mathbf{D_2} & \mathbf{D_3}\end{bmatrix}$ [51]. A graphical representation of this step is expressed in the diagram of figure 4.1.

After the preparation phase, one may assume the principal directions do not change for the upcoming respiratory cycles. However, since the needle-tissue interaction is expected to influence the fiducial marker positions it might be interesting to evaluate the instantaneous principal components' evolution during the needle insertion. The computational effort of this step is dependent of the size WP of the PCA Rolling Buffer.

After the first estimation of the principal directions of the marker positions, the prediction phase starts. Summarized in the diagram of figure 4.2, it starts by reducing the dimensionality of the marker position into it's first principal direction. The principal positions for each time step are stored in a single-dimensional rolling buffer of size WF. A linear regression is then performed using a first-order model matrix A_1 over all points in the buffer.

$$\begin{bmatrix} t_0 & 1 \\ t_1 & 1 \\ \vdots & \vdots \\ t_{WF} & 1 \end{bmatrix} \hat{\mathbf{x}} = \begin{bmatrix} P_C^{(0)} \\ P_C^{(1)} \\ \vdots \\ P_C^{(WF)} \end{bmatrix}$$

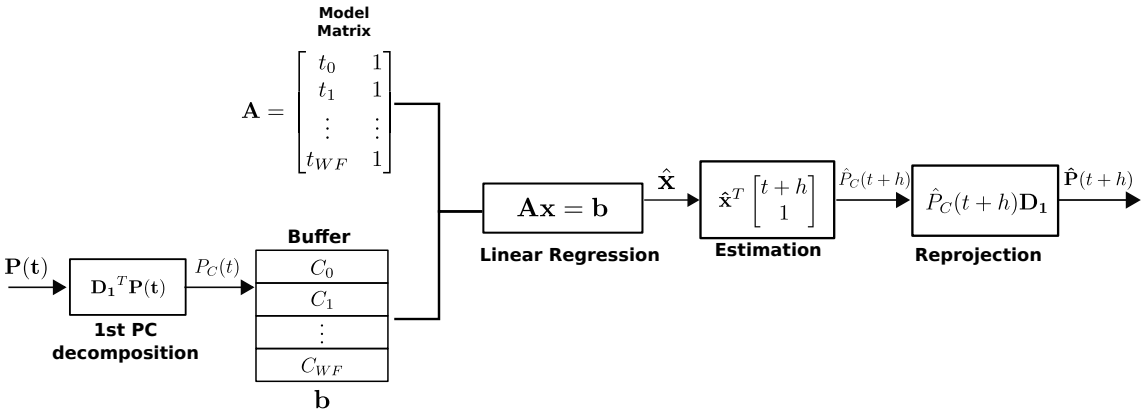

Figure 4.2: Prediction phase for a first order linear prediction filter. Each marker position ($\mathbf{P(t)}$) is decomposed into it's first PC ($P_C(t)$) and is stored into a single dimensional rolling buffer. A linear regression is then performed to fit the buffer values into a first order model matrix $\mathbf{A}$. As output, the filter weights $\hat{\mathbf{x}} \in \mathbb{R}^2$ are computed. Based in the filter weights $\hat{\mathbf{x}}$ a linear filter (LF) estimation is performed to estimate the principal component of the signal at timestep $t + h$, where h is a constant time-horizon. Finally, 3D estimates of the marker position are estimated by reprojecting the signal first principal component into the cartesian space $\mathbf{P}(\hat{t + h})$.

$$\mathbf{A_1}\hat{\mathbf{x}} = \mathbf{b} \tag{4.2}$$

The estimated prediction weights $\hat{\mathbf{x}} \in \mathbb{R}^2$ are then used to predict values for the PC in a future horizon h in equation 4.3.

$$\hat{P_C}(t + h) = \hat{\mathbf{x}}^T \begin{bmatrix} t + h \\ 1 \end{bmatrix} \tag{4.3}$$

Finally, the predicted value is reprojected into the Cartesian space in equation 4.4 and configures a predicted position of the fiducial marker position.

$$\hat{\mathbf{P}}(t + h) = \hat{P_C}(t + h)\mathbf{D_1} \tag{4.4}$$

4.1.2 | Higer Order LF Prediction

For distant time-horizons and large linear regression window WF it is possible that the first-order LF predictor presents elevated errors in specific periods. This is specially observed at the extreme point of a semi-periodic signal, such as the respiratory motion. On preliminary experiments, the instantaneous error could reach up to a millimeter from the original marker position.

An intuitive solution to this issue is to increase the order of the model matrix to include a second order polynomial terms. So the equation 4.2 can be rewritten as equation 4.5.

$$\begin{bmatrix} t_0^2 & t_0 & 1 \\ t_1^2 & t_1 & 1 \\ \vdots & \vdots & \vdots \\ t_{WF}^2 & t_{WF} & 1 \end{bmatrix} \hat{\mathbf{x}}_2 = \begin{bmatrix} P_C^{(0)} \\ P_C^{(1)} \\ \vdots \\ P_C^{(WF)} \end{bmatrix}$$

$$\mathbf{A_2}\hat{\mathbf{x}}_2 = \mathbf{b} \tag{4.5}$$

Similarly, the estimated prediction weights $\hat{\mathbf{x}}_2 \in \mathbb{R}^3$ are then used to predict values for the PC as in equation 4.6. Then, the reprojection equation remains the same as expressed in equation 4.4 with different weight vector dimension $\hat{\mathbf{x}}_2 \in \mathbb{R}^3$.

$$\hat{P_C}(t+h) = \hat{\mathbf{x}}_2^T \begin{bmatrix} (t+h)^2 \\ t+h \\ 1 \end{bmatrix} \tag{4.6}$$

4.1.3 | Horizon Estimation

According to Ernst et al. [50], a key aspect to perform respiratory motion compensation using prediction filters is to estimate the system delay we expect to compensate. Latencies occur naturally in the signal processing chain of a robotic system. To perform motion compensation, an estimate of the entire system delay must be used as horizon to the LF predictor. So one needs to add together the acquisition time to localize the fiducial markers $\Delta\tau_{loc}$ and the inverse simulation computation time $\Delta\tau_{invSimu}$. The final system delay would be close to the sum of all these instances, as in equation 4.7.

$$\tau = \Delta\tau_{loc} + \Delta\tau_{invSimu} \tag{4.7}$$

While the time required to extract and localize the fiducial markers $\Delta\tau_{loc}$ can be approximated to a constant time, $\Delta\tau_{invSimu}$ cannot. The time needed to solve the inverse simulation $\Delta\tau_{invSimu}$ is not deterministic and may vary significantly – first because our software run in a traditional (not a real-time) Linux operating system, and because the iterative solver may take a variate amount of time to reach a solution for each time step. From a preliminary observation in the profile of this delay, in figure 4.3, the most notable features are an always crescent, slow variation of the delay profile as a simulation is running. So, in order to estimate $\Delta\tau_{invSimu}$ we employed a second linear filter, designed to predict the next simulation loop timestamp.

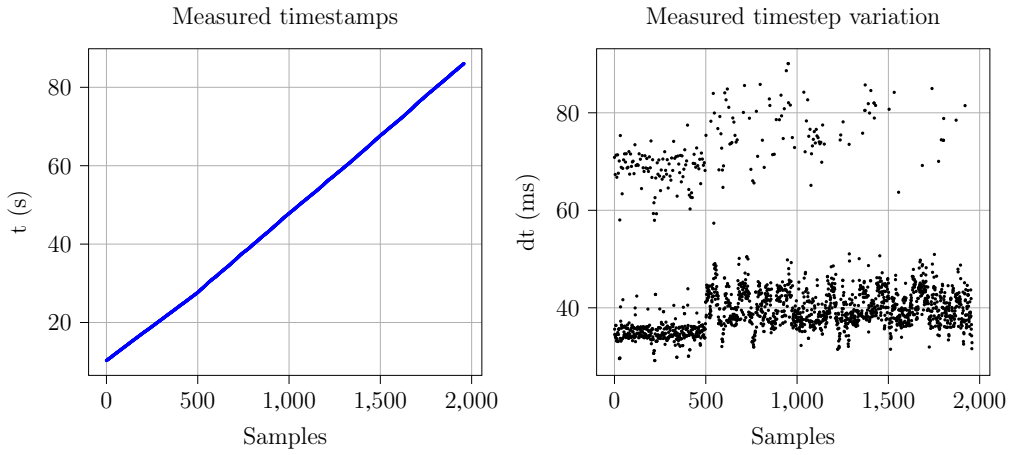

Figure 4.3: Timestamp measurements of a needle insertion simulation and their instantaneous variation.

The procedure is really similar to the one described to predict marker positions. This time, no dimensionality reduction is needed, since the signal only possesses a single dimension. The linear regression of equation 4.2 might easily be adapted to handle timestamps instead of PCs. The resulting expression is in equation 4.8.

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \\ \vdots & \vdots \\ WF & 1 \end{bmatrix} \hat{\mathbf{x}}_d = \begin{bmatrix} t^{(0)} \\ t^{(1)} \\ \vdots \\ t^{(WF)} \end{bmatrix}$$

$$\mathbf{A_{1d}}\hat{\mathbf{x}}_d = \mathbf{b}_t \tag{4.8}$$

Then, the future timestamp can be predicted with a simple adaptation of the equation 4.3.

$$\hat{t}^{(WF+1)} = \hat{\mathbf{x}}_d^T \begin{bmatrix} WF+1 \\ 1 \end{bmatrix} \tag{4.9}$$

So the prediction horizon which accounts for the computational time delay should be added with the expected delays from the camera acquisition and the robotic movement, as in equation 4.10.

$$\hat{h} = \hat{t}^{(WF+1)} + \Delta\tau_{loc} \tag{4.10}$$

4.1.4 | Hyperparameter Optimization

For a constant horizon, the only hyperparameter which influences the LF prediction is the size of the regression window $WF \geq n$ where n is the filter order. The same

goes to the horizon estimation explained in subsection 4.1.3. It is possible to set this property manually. A small regression window would improve the filter reactivity and potentially improve its precision. A large regression window would provide a relative robustness to measurement noise, while sacrificing the filter reactivity.

An alternative would be to implement a uni-dimensional search method to find the regression window which minimizes the prediction error. One can take advantage of the prediction phase, expressed in figure 4.2, as training data to optimize the hyperparameter WF. Having the size of the PCA buffer typically way larger than the regression window $WF \ll WP$, one can iterate through the PCA buffer trying to predict its next value. A first step to implement this optimization is to propose the cost function $C(WF)$ we intend to minimize.

Algorithm 1 details how one can use the PCA buffer to obtain a quadratic cost for a given regression window WF. The function call $X.getPrincipalComponent()$ expresses the projection step of figure 4.1 over the entire buffer X. A loop over the principal components of the buffer PC attempts to predict its next data using the linear filter. The function $PC.toModelMatrix()$ outputs a model matrix A in the form expressed in equation 4.2. Then calling $predict(next\_time)$ is equivalent of computing the equation 4.3 with $h = next\_time$. Finally, the algorithm outputs the mean quadratic error as the cost for the given prediction window WF.

Then, as unidimensional optimization method the golden section method was chosen to find a local minimum $\hat{WF}$ [52]. First 4 points are elected as candidates to a local minimum : $p_1, p_2, p_3, p_4 \in \mathbb{N} \in [n, WL-1]$. As an iterative solution, at time step 0 $p_1^{(0)} = n$ is first assigned to the minimum of the cost function search domain and $p_4^{(0)} = WL - 1$ is assigned to it's maximum. The two intermediate variables are computed according to the golder ratio proportion $\phi = \frac{\sqrt{5}+1}{2} \simeq 1.618$ and is expressed in equation 4.11.

$$p_2^{(0)} = p_1(0) + \frac{p_4(0) - p_1(0)}{\phi} \tag{4.11}$$

$$p_3(0) = p_4(0) - \frac{p_4(0) - p_1(0)}{\phi} \tag{4.12}$$

The cost function is evaluated over each point and their values is stored into intermediate variables $C_i = C(round(p_i)), i \in [1, \ldots, 4]$. As $WF \in \mathbb{N}$, the method was adapted to always round the inputs for the cost function evaluation. The update routine for the following iteration will depend on which intermediate variable $C_2^{(k)}$ or $C_3^{(k)}$ for an arbitrary iteration $k > 0$ is the smallest.

- Case $C_2^{(k)} < C_3^{(k)}$: local minima is most probably in the interval $\in [p_1^{(k)}, p_3^{(k)}]$

and their values are kept, while $p_4^{(k)}$ changes.

$$p_1^{(k+1)} = p_1^{(k)} \tag{4.13}$$
$$p_4^{(k+1)} = p_3^{(k)} \tag{4.14}$$

- Case $C_2^{(k)} > C_3^{(k)}$: local minima is most probably in the interval $\in [p_2^{(k)}, p_4^{(k)}]$ and their values are kept, while $p_1^{(k)}$ changes.

$$p_1^{(k+1)} = p_2^{(k)} \tag{4.15}$$
$$p_4^{(k+1)} = p_4^{(k)} \tag{4.16}$$

In both cases p_2 and p_3 are updated according to the equation 4.17, which is really similar to 4.11.

$$p_2^{(k+1)} = p_1(k+1) + \frac{p_4(k+1) - p_1(k+1)}{\phi} \tag{4.17}$$
$$p_3^{(k+1)} = p_4(k+1) - \frac{p_4(k+1) - p_1(k+1)}{\phi} \tag{4.18}$$

Algorithm 1: Algorithmic representation of the cost function $C(WF)$ applied to the PCA buffer during the preparation phase.

```
DATA X: PCA buffer of size 3 × WP;
DATA A: regression buffer of size WF;
RETURNS cost: prediction quadratic cost for a given WF;
PC = X.getPrincipalComponent();
cost = 0;
counter = 0;
N = WP − WF − 1;
while counter < N do
    A = PC.toModelMatrix(counter, couter + WF);
    next_val = PC[counter + WF + 1].value();
    next_time = PC[counter + WF + 1].time();
    prediction = A.predict(next_time);
    cost += (prediction−next)²/N;
    counter += 1;
return cost;
```

4.1.5 | Performance Metrics

In order to evaluate the filter performance, the same metrics used by Johl et al. can be explored. They propose using the Root Mean Square Metrics (RMS) and

Normalized Root Mean Square (nRMS) comparing the predicted PC of the signal and it's measured PC. After uniformly resampling the input signal $P_C^{(i)}$, $i = 1, \ldots, N$ and $\hat{P_C}^{(i)}$ the output of the prediction filter, the RMS error is computed as in equation 4.19.

$$e_{RMS}(\hat{P_C}, P_C)^{(i)} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} ||\hat{P_C}^{(i)} - P_C^{(i)}||^2} \tag{4.19}$$

The nRMS error represents the amount of improvement a prediction can achieve over the prediction horizon δ. Its mathematical expression is the ratio between the predicted signal RMS and the signal delayed by the prediction horizon ($D(y, \sigma)$) as expressed in equation 4.20. As an example, if the nRMS is equal to 75% it means the prediction algorithm reduces the RMS error by 25% when compared to doing no prediction [53].

$$e_{nRMS}(\hat{P_C}, P_C, \sigma)^{(i)} = \frac{\text{RMSE}(\hat{P_{C\sigma}}, P_C)^{(i)}}{\text{RMSE}(D(P_C, \sigma), P_C)} \tag{4.20}$$

4.1.6 | Motion Compensation Strategy

In a traditional inverse simulation, the liver biomechanical model is constrained by the registered position of the fiducial markers attached to the patient's liver. An alternative to enhance the inverse simulation with predictive information about the system is to combine it with the delay prediction and the prediction models for respiratory motion. While an instance of the linear filter can be used to estimate the future horizon $\hat{h}$, another can predict each fiducial marker position at this next inverse simulation timestamp. This behavior is expressed in contrast with the traditional inverse simulation in figure 4.4.

4.2 | Beveled tip needle FEM

A first contribution of this master thesis is to enhance the FEM of needle insertion described in section 3.1 to emulate the behavior of asymmetric beveled tip needles. The addition of another DoF in the needle tip can improve the reactivity of the needle insertion framework by inverse simulation. A first desired feature for this model is to be compatible with the needle insertion framework used in the team. Also, the model should have measurable parameters to fit in a real needle insertion data in phantoms. The bevel needle model parameters are fitted to reproduce gelatin phantom experiments results informed by fellow researchers at University of Verona.

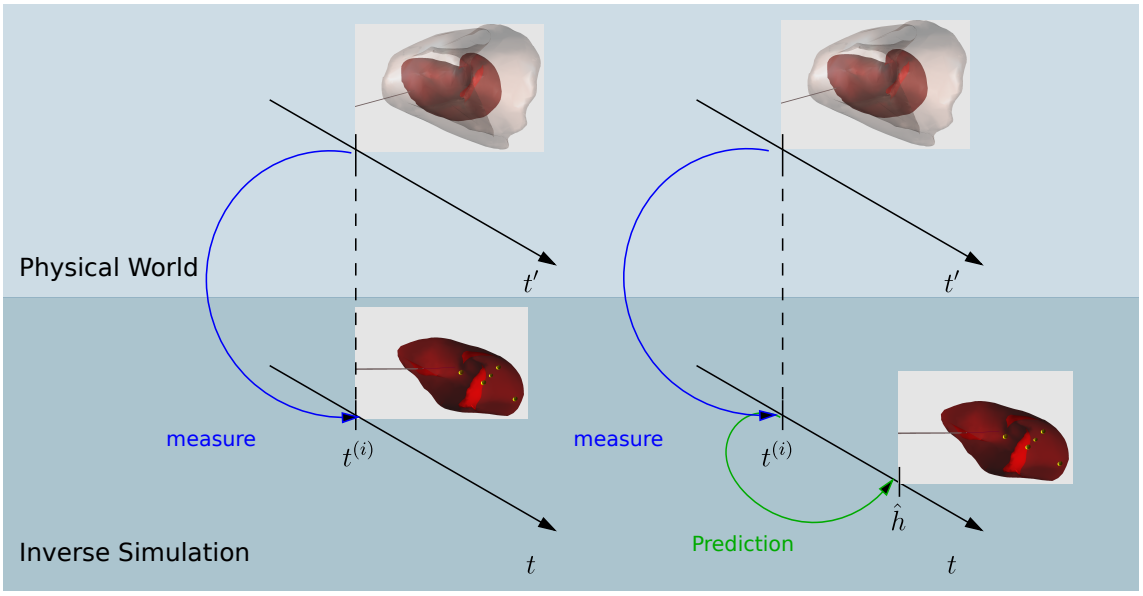

Figure 4.4: Predictive inverse simulation workflow compared to the traditional inverse simulation. On the left side, the traditional inverse simulation step uses the measured fiducial marker position to register the biomechanical model of the inverse simulation. On the right, the predictive inverse simulation uses the measured position as inputs to the motion prediction filter. It's the prediction output for a time horizon h in the future which is used to register the inverse simulation with respect to the physical world. t' denotes the continuous time of the physical world.

4.2.1 | Misra model for beveled tip needles

As discussed in the state-of-the-art 2.1, many models to represent the asymmetric needle deflections exist in the literature. While both models based on the needle kinematic or mechanical energy formulation could be integrated into the inverse simulation, the mechanical energy fits best as an addition to the FEM needle insertion model used in the team [17]. Misra's needle deflection mechanism only rely on a force acting on the tip, which can be easily added to the existing needle-tissue interaction model. So this study will focus in implementing a SOFA compatible version of Misra's beveled tip needle model.

In order to fit this model to experimental data, only the needle radius and it's stiffness were informed. Also, an estimate of the needle-tissue interaction stiffness K_T. The experimental data should be fitted by tuning the unknown parameter: the ratio between the cut angle β and the bevel angle α.

4.2.2 | Model as a ForceField

In order to integrate Misra's needle tissue interaction model represented in equations 2.3 to 2.1 into SOFA, a new *ForceField* component was created. As the model's forces and torque do not depend on the needle position, additional lines can be added to the internal force matrix $\mathbb{F}$ in equation 3.1. So the model forces represented in figure 2.1 will act as constant internal forces applied to the needle tip whenever inside the tissue.

In order to correctly implement this beveled needle model as a *ForceField* it is also important to choose whether to apply the transversal force Q or the resultant torque M. Imposing to the FEM simulation both would mean the implicit integration scheme would account twice for its effect inside the simulation. During this study the application of the torque M was privileged since the force Q presented instabilities at the insertion point.

A last step is to register these forces and torques with respect to the needle tip in it's three-dimensional coordinates. The needle tip position is extracted from the simulation and the forces are registered to it according to the tip frame expressed in figure 2.1.

4.2.3 | Integration to the inverse simulation

The usage of a asymmetric needle in needle steering procedures is common in the literature. As an example, Chevrie et al. use needle base manipulation to steer their needles in soft tissues [43]. The bevel model corresponds to an additional degree of freedom in the needle tip. This can enhance the final precision of the needle insertion procedure, since the control framework will dispose of another resource to compensate the physiological motion.

5 | Results and Discussion

All results presented in this study were obtained with system measures of SOFA C++ components implemented by the team Sperry. The main computer used for result acquisition has an Intel® Core™ i7-4790K CPU @ 4.00GHz × 8 processor, with 16 GB RAM and graphics card NVIDIA Corporation GM204 [GeForce GTX 970]. An auxiliary computer was used to run the direct simulation described in 5.1 with similar configuration: Intel® Core™ i7-9750H CPU @ 2.60GHz × 12 processor, 16 GB RAM and graphics card NVIDIA Corporation TU106M [GeForce RTX 2060 Mobile].

5.1 | Realistic Simulations Setup

First the performance of the prediction filters will be validated with respect to the performance reported by Johl [38]. The input marker positions will have uniform white Gaussian noise added to their measurements in order to evaluate the filter robustness to different signal to noise ratios (SNR). Also, the filters will be evaluated under different magnitudes of the prediction horizon. For each horizon, the regression window hyperparameter optimization strategy will also be validated.

In a second moment, in order to validate the motion compensation aspect of the prediction filters, an experimental setup similar to Baksic's is going to be implemented [2]. The motion compensation aspect of his solution was validated using two parallel biomechanical simulations. A direct biomechanical simulation, which is as realistic as possible and attempts to replicate the real world, and an inverse simulation as described in section 3.2.

The direct simulation takes advantage of the preoperatory phase in figure 1.1 to model not only the patient's liver, but also it's skin and it's rib-cage. All these components are modeled into the direct simulation scenario. To replicate the anatomical effect of the respiratory motion, a realistic semi-periodic motion is applied to an upper bounding box of the liver mesh. This is supposed to emulate the diaphragm imposing internal movement to the liver. Two of the ribs are also modeled in contact with the liver, constraining its movement inside the patient's body. All these interactions can be viewed graphically in figure 5.1. The inverse simulation and

prediction filters are unaware of these interactions.

The liver biomechanical model used is a tetrahedral mesh composed of 2660 nodes with Young Modulus $5.5kPa$ a Poisson ratio of 0.45. The needle model used in section 5.2 is a rigid needle with $130cm$ length, $0.723mm$ radius, $200GPa$ Young Modulus and Poisson ratio 0.45. In section 5.3, a flexible needle is used to account best for the bevel needle model deflection: $0.6mm$ radius, $5GPa$ Young Modulus and Poisson ratio of 0.45.

As both simulations are tuned to run at a fast rate in a single CPU thread and multiple GPU cores it cannot be run alongside with the controller scene with the inverse simulation. The busy access to the CPU drops significantly the performance of both simulations. So a solution is to run the two simulations into two different machines, connected by a UDP bus.

In the first computer, the inverse simulation described in section 3.3. When the needle is outside the tissue it can operate at approximately $30Hz$ and drops to about $20Hz$ when the needle is inside the liver. The mean rate for the inverse simulation steps is $26.96Hz$. In the second computer, the direct simulation will emulate the physical world. We were able to measure up to $35Hz$ while the needle is outside the tissue, around $28Hz$ when the needle is in between the skin and the liver and about $20Hz$ when the needle is inside the liver. The mean value for the entire simulation was $28.5Hz$.

With respect to the system delays, as the marker positions at an arbitrary instant i, $\mathbf{m^{(i)}}$, will be the actual mesh positions streamed through the UDP network, the marker detection delay can be approximated to zero $\Delta\tau_{loc} \rightarrow 0$. Also, the additional delay of the network connection is of the order of microseconds and should not interfere much with the horizon expression of equation 4.10.

5.2 | Respiratory motion prediction filters

Respiratory prediction filters were first validated by their performance in a free-breathing scenario – without the needle disturbances in the tissue. Then their performance was evaluated during the needle insertion. First to predict the delay between timestamps of the inverse simulation and then to predict markers position. Finally, the results of a predictive simulation which uses the LF as a prediction model for the next time step are compared to the traditional inverse simulation results.

5.2.1 | Free Breathing Prediction Performance

Throughout this section the prediction filters will be evaluated in a digital signal processing perspective. Previously acquired internal fiducial marker positions were used as input to the filter with different levels of Gaussian noise added to them. Figures 5.3 and 5.4 expresses a quantitative overview of the filter performance met-

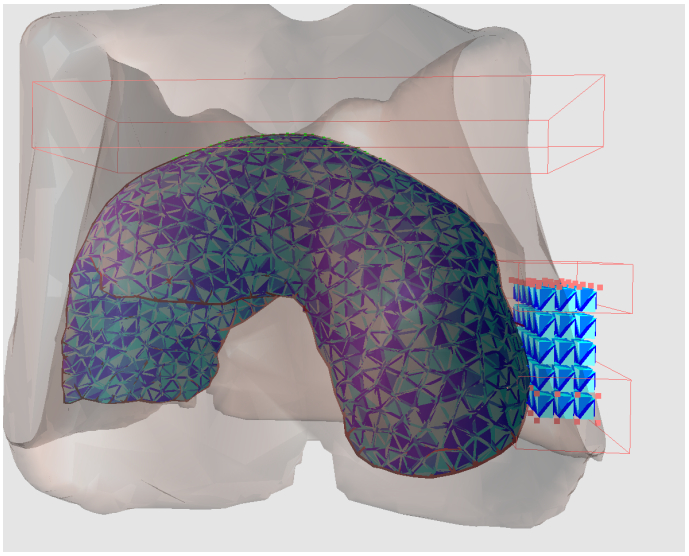

Figure 5.1: Detailed view of the biomechanical models involved in the direct simulation. The liver is modeled into a tetrahedral mesh with 2660 nodes and the skin between ribs with 144 nodes. The two lateral bounding boxes represent the rib-cage and will constrain the insertion point movement. The superior bounding box represents the diaphragm and the respiratory motion is induced only in the liver node inside the box.

rics for different input SNR. The same overview is also applied to the filter after regression window hyperparameter optimization in figures 5.5 and 5.6.

It's possible to observe in figures 5.3 to 5.6 the RMS Error precision is both affected by the increasing horizon of prediction and by noise level. Although in every most cases the final RMS error is inferior to $1mm$, the normalized RMS error (nRMS) metric expresses the filter is most effective in low SNR levels when compared to not doing prediction at all. It's possible to conclude the second order filter is even more sensitive to noise, since it presents both RMS errors above $1mm$ and nRMS well beyond 100%.

When comparing figures 5.3 and 5.4 to figures 5.5 and 5.6 it's possible to see a precision improvement for filters of both orders due to the regression window optimization. Both the absolute RMS value and the nRMS are inferior to the fixed regression window case to every SNR and every horizon. For low levels of the SNR, however, even after the regression window optimization in figures 5.5 and 5.6, the $nRMS$ remains close to 100% for most of the markers. This agrees to Johl et al.'s observation that the smoothness of the input signal is determinant for the filter's performance.

As a conclusion, filters of both orders presented substantial improvements when coupled with the regression window optimization routine described in 4.1.4. For high SNR levels, the second order prediction filter presented a better performance than

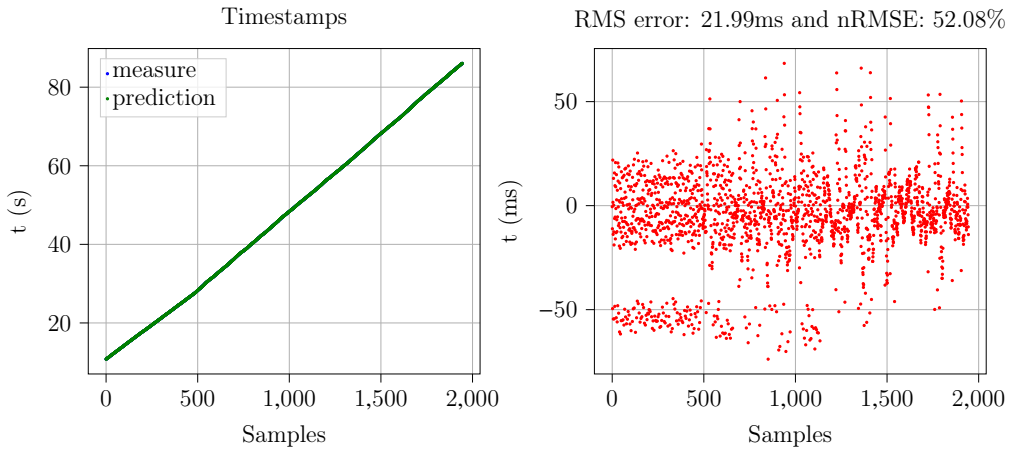

Figure 5.2: Immediate future timestamp prediction using a linear filter.

the first order. It becomes more and more sensitive to noise as the SNR decreases. In future experiments, marker position prediction will be performed using a second order linear prediction. This will be the case for the motion compensation strategy performed in section

5.2.2 | Delay Prediction

As proposed in section 4.1.3, the computation time a linear filter was used to predict the computation time $\hat{t}^{(WF+1)}$. A first order model matrix was used, since only a low frequency variation is expected of the signal of figure 4.3. The hyperparameter optimization strategy was used to compute the filter regression window.

The prediction result is expressed in figure 5.2. Even though the left side of the figure does not inform much on the filter performance, the error plot on the right side expresses a final RMS and nRMS error measurements. Even though the prediction error itself is elevated, the normalized metric reveals there's almost a 50% improvement in using the filter to predict the next timestamp. As expected, the high frequency variations of the time step expressed on the right side of figure 4.3 could not be predicted by the prediction filter and remain present in the error plot of figure 5.2

5.2.3 | Prediction Performance During Needle Insertion

A first result when attempting to achieve motion compensation is to evaluate whether the linear prediction filter is effective when used in a needle insertion scenario. Up to this point the prediction filter has been validated in respiratory simulated data of free breathing, without interaction between needle and tissue. Figure 5.7 expresses a single view of the prediction filter performance during a simulated needle insertion procedure.

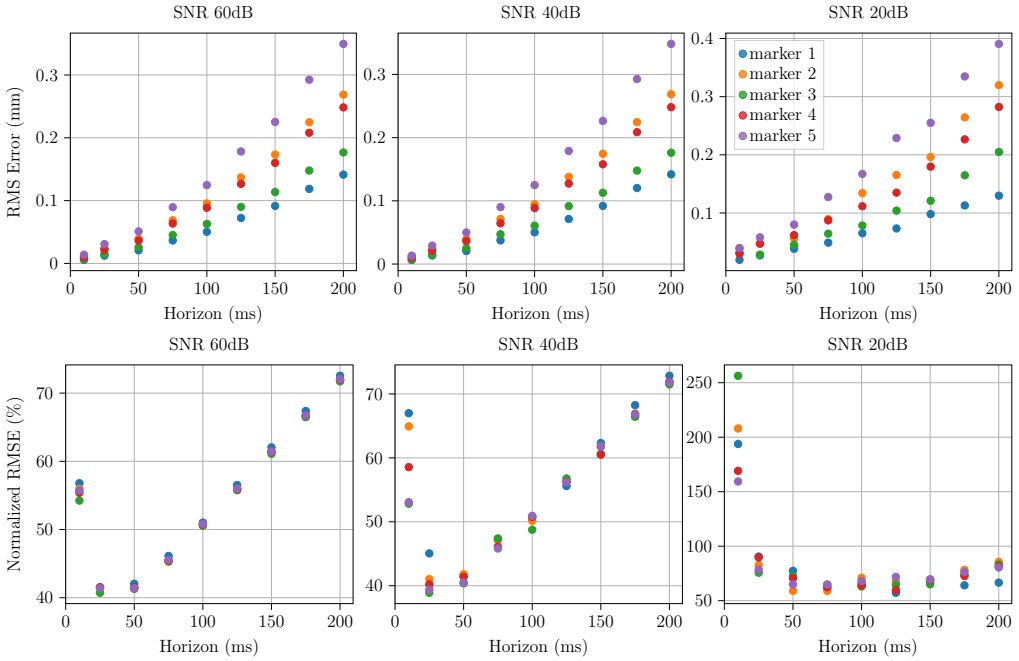

Figure 5.3: First order Linear prediction filter performance over simulation data for 60*dB*, 40*dB* and 20*dB* SNR. A fixed regression window of 10 samples was used for both filters.

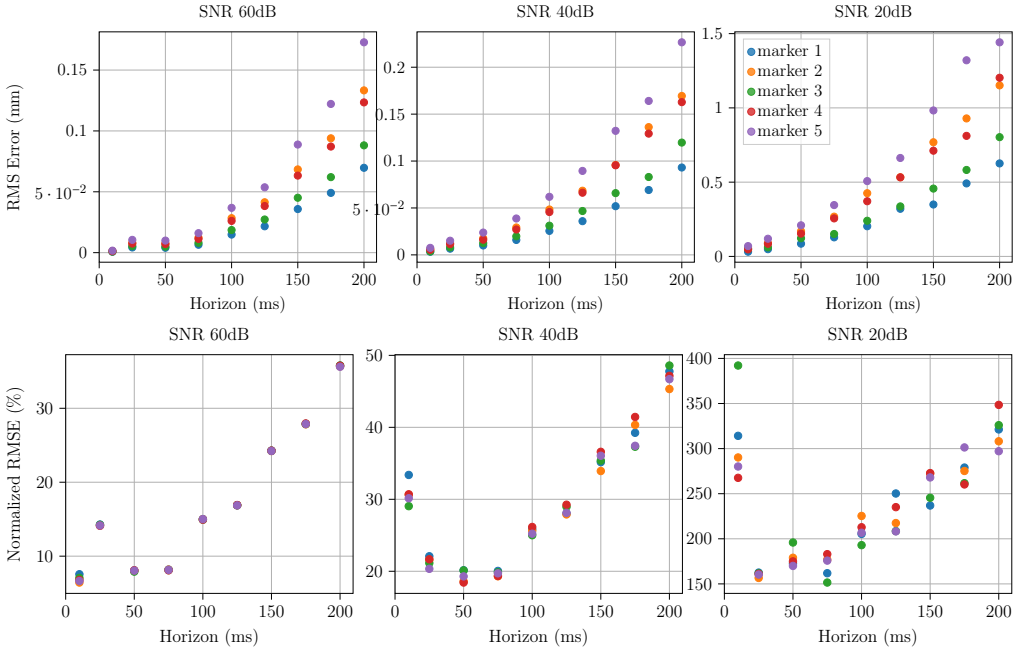

Figure 5.4: Second order Linear prediction filter performance over simulation data for 60*dB*, 40*dB* and 20*dB* SNR. A fixed regression window of 10 samples was used for both filters.

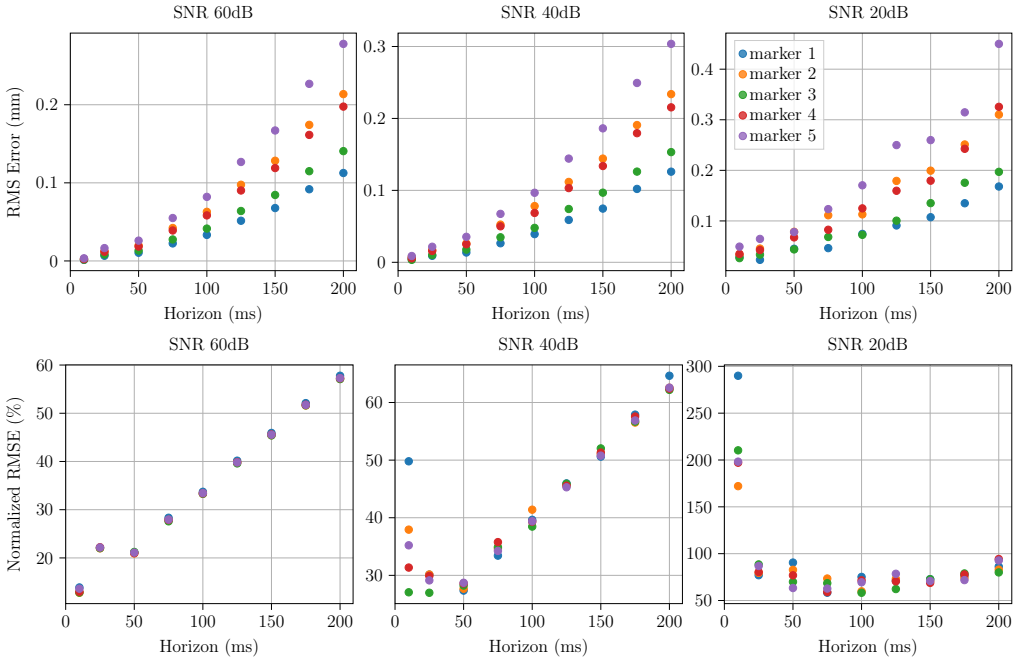

Figure 5.5: First order Linear prediction filter performance over simulation data for 60*dB*, 40*dB* and 20*dB* SNR. An optimal regression window was used for each signal as described in section 4.1.4.

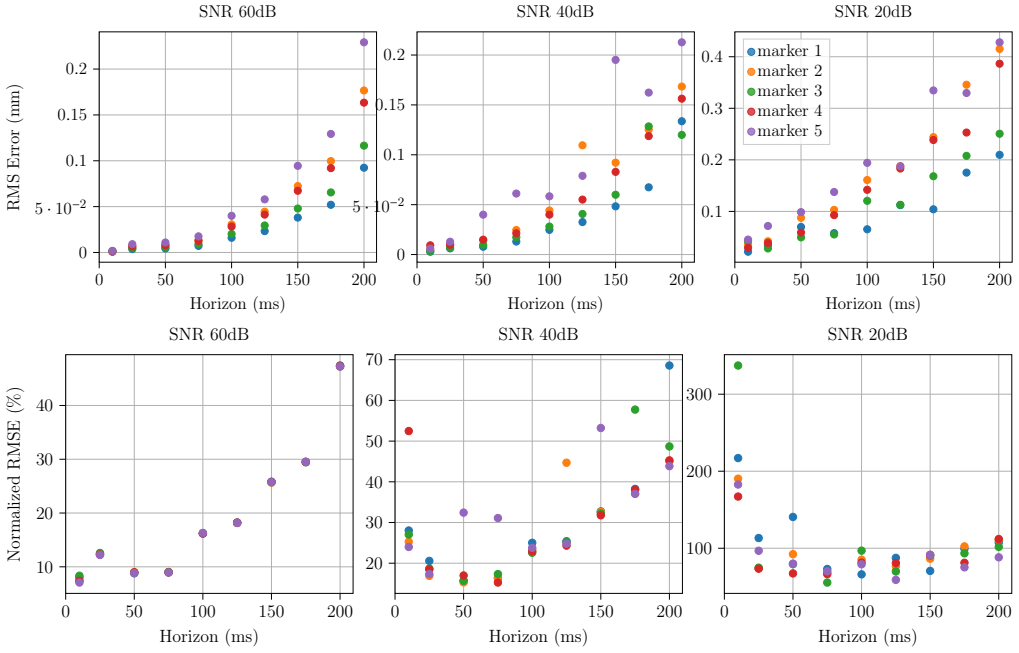

Figure 5.6: Second order Linear prediction filter performance over simulation data for 60*dB*, 40*dB* and 20*dB* SNR. An optimal regression window was used for each signal as described in section 4.1.4.

Among the other markers, these performance metrics presented some variation as expressed in table 5.1. While the RMS error was observed to stay stable and well below a millimeter, the nRMS error surpassed the 100% threshold for one of the markers. Figure 5.7 presents the worst prediction case in this sample. Although the final RMS error is well under a millimeter and the nRMS metric reveals the filter is unable to improve the delayed version of the signal. As table 5.1 shows, this is not the case for all remaining markers. The LF prediction filter is actually able to improve in 70% marker 5's prediction.

Table 5.1: LF prediction performance for each marker during a needle insertion scenario.

$marker$	1	2	3	4	5
$e_{RMS}(mm)$	0.066	0.044	0.045	0.033	0.035
$e_{nRMS}(\%)$	110.95	69.89	65.44	33.45	29.94

If one assumes the simulation to be noise-free – ignoring the numerical noise of the FEM – the elevated nRMS for marker 1 can be a consequence of the high frequencies observed in the marker PC in figure 5.7 towards the end of the needle insertion. These frequencies are a consequence of the needle and tissue interaction which was not accounted for during the first experiments of section 5.2.1. These interferences are also not considered during the preparation phase in which the regression window is optimized. So it is possible a different regression window could work best to predict the next positions along the needle insertion. Periodic runs of the hyperparameter optimization routine could be an interesting approach to explore in future applications.

5.2.4 | Motion Compensation Strategy

Finally, to fully integrate the prediction filters into the inverse simulation, the liver biomechanical model will be constrained to the predicted marker positions instead of the measured ones. In this setup, the inverse simulation will operate in a predicted state of the needle insertion scene and will generate robotic commands capable of compensating the liver motion in between the simulation timestamps.

First a second evaluation of the filter prediction performance is acquired. This time, the filter predictions will effectively influence the inverse simulation and the robotic commands. So in a similar experiment to the one described in section 5.2.3, the prediction performance of the filter was evaluated while attempting motion compensation. The results for a single marker are expressed in figure 5.8 and for each marker is expressed in table 5.2.

Once again, the first marker presents the worst nRMS case among all. This time, however, it's nRMS metric is below 100%, meaning the overall usage of the filter,

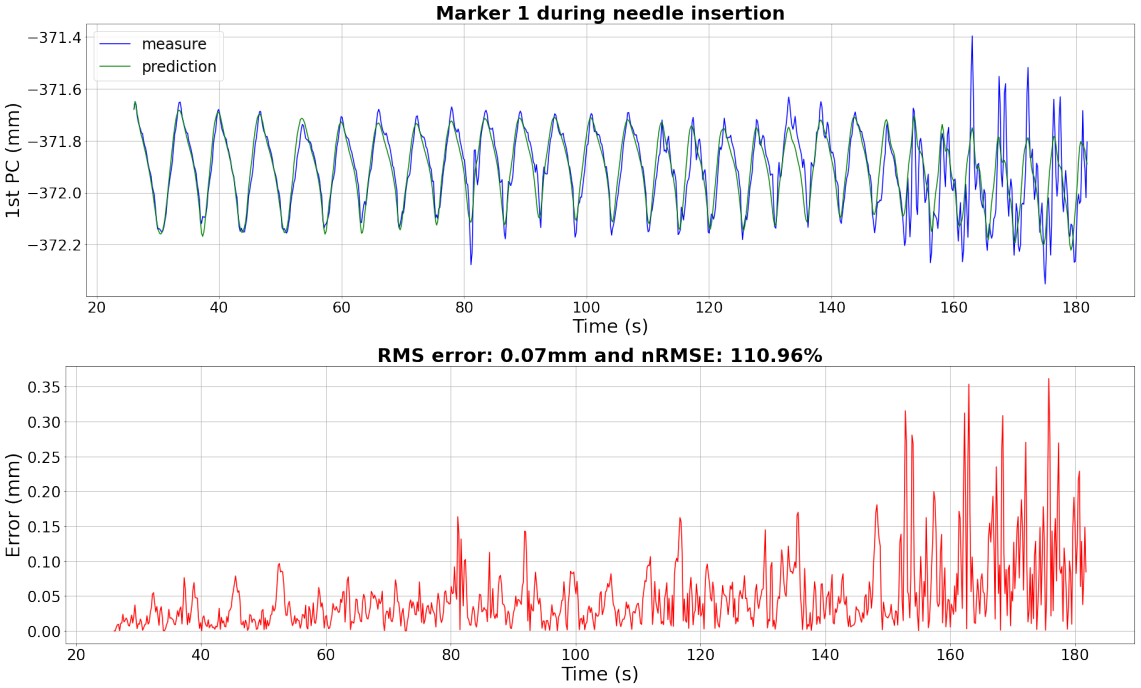

Figure 5.7: LF prediction metrics applied to a single marker in a needle insertion scene. The red curve is the error plot between measurement and prediction for a same timestamp after resampling.

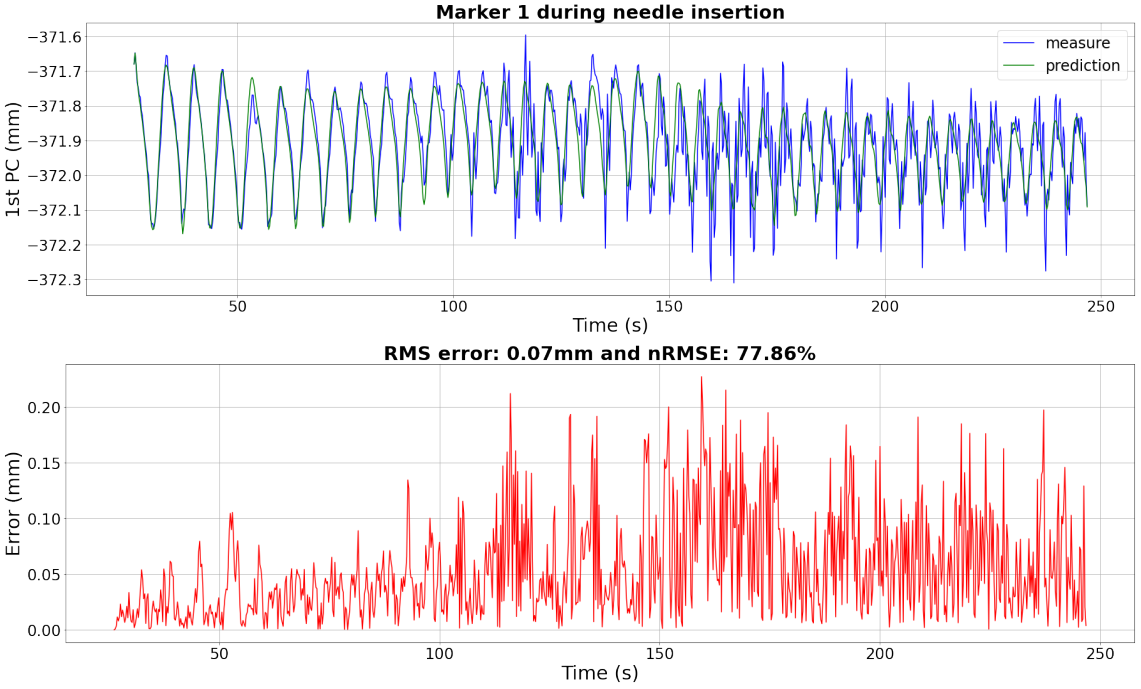

Figure 5.8: LF prediction metrics applied to a single marker while attempting motion compensation using the LF. The red curve is the error plot between measurement and prediction for a same timestamp after resampling.

Table 5.2: LF prediction performance while being used for motion compensation in the inverse simulation.

$marker$	1	2	3	4	5
$e_{RMS}(mm)$	0.067	0.047	0.043	0.0346	0.044
$e_{nRMS}(\%)$	77.85	51.88	45.79	24.46	27.25

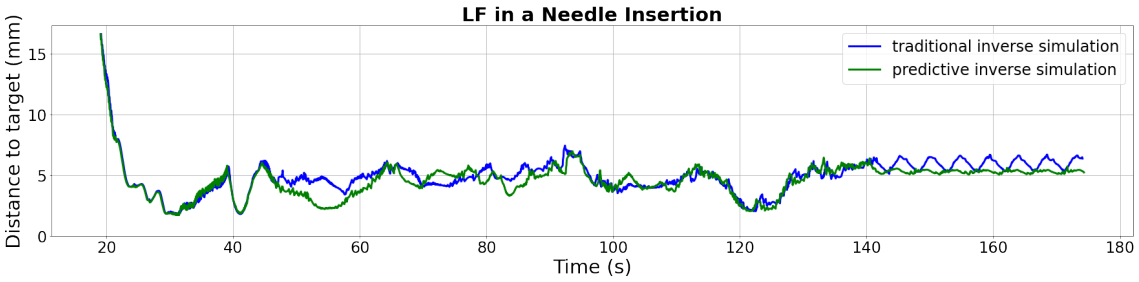

Figure 5.9: Motion compensation distance to target comparison between predictive inverse simulation and traditional inverse simulation. This corresponds to the objective function $\mathbf{e}_p$ of the inverse simulation control framework. This metric was measured from the direct simulation with a rigid needle with $200GPa$ Young Modulus.

even with motion compensation, improved the position error in 32% when compared to not using the filter at all. As all markers RMS metric remained stable and well below a millimeter and nRMS metrics below 100%, it is expected the prediction filter will improve the final needle insertion procedure.

Figure 5.9 expresses a comparison between the predictive inverse simulation and the traditional one. As a performance metrics, the immediate distance from the needle to it's next target is used. This distance corresponds to the first *objective constraint function* $\mathbf{e}_p$ and it evolves over time as the needle reaches a small enough distance from a target and moves on to the next one in the trajectory. The final needle position error is expressed in the final measurements of $\mathbf{e}_p$, from approximately $145s$ onwards.

In their steady state, while the traditional inverse simulation final position error oscillates around a mean value of $\bar{e}_t = 5.97mm$ with an amplitude $e_t = 1.41mm$, the predictive inverse simulation has a mean error of $\bar{e}_p = 5.28mm$ and amplitude $e_p = 0.68mm$. This represents a mean value reduction of 11.49% and amplitude attenuation of 51.61%.

5.2.5 | Discussion and limitations

The promising results of section 5.2.4 indicates the addition of a predictive component in the inverse simulation to perform needle insertions can improve its final position accuracy and reduce oscillations. This presents itself as an interesting perspective for percutaneous procedures in which the needle needs to be accurately positioned for a long period while the patient breathes. This is the case, for example of needle guided ablations or cryotherapies. It is important, however, to address this study limitations which need to be addressed in future works.

First, the assumption that prediction filters used in the robotic radiotherapy field of research can be suited for needle insertion procedures is not yet validated. This is an evident conclusion of the variate nRMS performance to predict the high frequency disturbances caused by the needle and tissue interaction observed in figure 5.7 and table 5.1.

In the specific case of the linear filter this issue can be tackled in future results by a different hyperparameter optimization strategy. This study attempted to use the linear filter preparation phase of figure 4.1 – in which the PC of the signals are computed – as a training phase to find the best regression window for each marker prediction filter. Even though this approach worked in free breathing cases explored in section 5.2.1, during needle insertions high frequency disturbances caused by the needle can undermine the filter performance.

A different hyperparameter optimization strategy should also be considered for future works. The golden section method described in section 4.1.4 assumes a convexity of the cost function described by the algorithm 1 which has not been validated. It is possible the cost function presents multiple local minimum which would not return the regression window for the training data. It is possible the Bayesian Optimization [1] solution used by Johl et al. can overcome this issue and find a guaranteed global minimum for each hyperparameter. In addition to that, a future optimization strategy could run alongside with the needle insertion procedure in order to adapt its hyperparameters to the high frequencies imposed by the needle into the tissue.

It would also be important in future studies to evaluate the motion compensation results in multiple simulations to address the robustness of the prediction filter to variabilities in the simulation biomechanical parameters. Baksic et al. has addressed this issue by adding a 13% variability in the young modulus as this is the average repeatability reported in a liver elastography survey [54]. So in their final needle position error results, they've presented an average over 22 simulations with different mechanical properties.

Another important observation is that, even though this study was developed in the same laboratory and with Baksic's solution accessible, the results found using their method were different from the ones reported in their last work [2]. While the final needle position average found in this study using their method was $5.97mm \pm$

[1] https://github.com/fmfn/BayesianOptimization

1.41, they've reported a final positioning error of $1.08mm \pm 0.44$ in their work. In addition to that, the evolution in time of the distance to target observed in figure 4.4 oscilates much before stabilizing at the final error measurement. This is probably a consequence of recent major changes in the source code of the simulations and the different computers used in order to obtain the results. It would be interesting to reevaluate the prediction filter influence as soon as the reported performance of the simulations is recovered.

5.3 | Beveled tip needle finite element model

The beveled tip needle FEM model enhancement will be tested in two routines. First a test scenario recreates an experimental measure acquired by researchers at University of Verona. This first study's objective is to fit our bevel needle *ForceField* to reproduce the final needle deflections observed through the gelatin. In a second experiment, the FEM model will be integrated into the realistic needle insertion simulation described in section 5.1. The final needle position error will be evaluated for a scene of the inverse simulation with the bevel needle model and without it.

5.3.1 | Fit beveled tip deflection to a phantom experiment

A graphical representation of this experimental setup is expressed in figure 5.10. The scenario consists in a virtual gelatin prism positioned right in front of a needle. Gelatin is modeled as a $30 \times 15 \times 5$ node mesh with tetrahedral elements and a co-rotational deformation model. The young modulus of the gelatin is set to $5GPa$. A bevel tip needle FEM has $20cm$ length, $d = 1.2mm$ diameter and unknown bevel angle. The young modulus of the needle is set to $43.26KPa$ and its bevel angle was informed as $\alpha = 0.2617rad$. The Poisson coefficient for both bodies is set as 0.45. A first estimate of the needle-tissue stiffness interaction was also informed as $150KPa$ A movement is imposed to the needle shaft along the **X** direction, inserting the needle it right in the center of one face of the gelatin block. It is expected to observe a deflection of $10mm$ in the **Y** direction at the end of the needle insertion.

Another parameter which also plays a role in the final needle deflection is the distance between needle constraints Δl_c applied to the tissue [17]. This corresponds to a discretization non-linearity of the needle insertion model constraint points inside the tissue.

It was possible to manually tune the beveled tip needle model parameters to fit the needle deflection observed in their experimental data. The model parameters and the model forces and torque outputs computed are expressed in table 5.3. These values were obtained maintaining the first estimate of K_T while varying β according to the common $\frac{\beta}{\alpha}$ values reported by Misra et al. The β parameter found to fit the experimental data correspond to a $\frac{\beta}{\alpha} = 0.9$. The constraint distance between constraints of the FEM of needle insertion was $\Delta l_c = 3.5cm$ A final position error

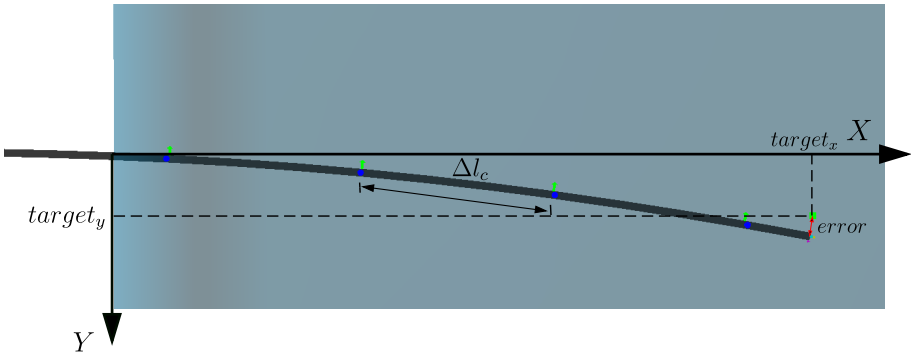

Figure 5.10: Graphical representation of the beveled tip needle scenario. The experimental measure of a target position acquired by them is $(0.31mm, 0.01mm)$ at the X/Y plane. This study will manually tune the model parameters to achieve the appropriate needle deflection at the end of the needle insertion. The most important parameters to tune is the ratio between needle advance angle and bevel angle β/α. Also the distance between needle constraints Δl_c seems to play a role in the final needle deflection. This results was acquired with $K_T = 150KPa$,$\Delta l_c = 3cm$ and $\beta/\alpha = 1.2$.

$0.14mm$ with the described model parameters.

Table 5.3: Bevel needle model parameters which fit the phantom experimental results.

Parameter	K_T	β	M
	$150KPa$	$0.236rad$	$0.48mN \cdot m$

5.3.2 | Integration to the inverse simulation

The bevel needle model is applied to both the direct and inverse simulation to evaluate if the control framework is able to command asymmetric needles. Similar to section 5.2.4, the immediate distance from the needle to it's next target during a needle insertion will be the performance metrics evaluated. A flexible needle model with Young Modulus of $5.5kPa$ will be used along this experiment. The same needle geometry described in section 5.3.1 will be used. As needle-tissue stiffness, the same value of $K_T = 150KPa$ was used for both the skin and liver needle-tissue interaction stiffness. This is an approximation and a precise estimate of the needle-tissue interaction of these tissues should be used in future experiments.

The results of the beveled tip needle model integrated to the inverse simulation is expressed in figure 5.11. Once again the distance to the immediate target $\mathbf{e}_p$ is used as a comparison metrics between a traditional inverse simulation and a beveled tip

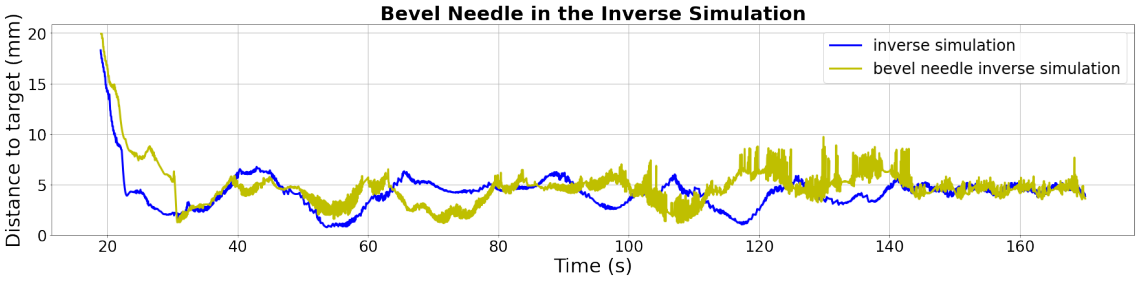

Figure 5.11: Comparison between beveled tip needle model integrated into the inverse simulation and traditional inverse simulation. The distance to target was used as a performance metric between both frameworks. A symmetric needle was used in both direct and inverse simulation of the blue curve. This corresponds to the objective function $\mathbf{e}_p$ of the inverse simulation control framework. This metric was measured from the direct simulation with a flexible needle with $5.5kPa$ Young Modulus.

inverse simulation. The blue results express a traditional inverse simulation needle insertion with a flexible needle. The yellow uses the beveled tip needle model for both the direct and inverse simulations.

An evident difference between the measured $\mathbf{e}_p$ in figure 5.11 is the presence of high frequencies in the bevel needle inverse simulation results. This represents the robot reaction to the beveled needle model torque imposed to the tissue around the needle tip position. It seems a larger amplitude of movements were commanded to the robot to compensate the needle deflection caused by the beveled tip model.

Both inverse simulations are able to reach a similar final mean position error value of $4.53mm$. While the amplitude of oscillations of the traditional remained around $1.5mm$, the beveled tip inverse simulation high frequencies oscillations presented $4mm$ instantaneous overshot.

5.3.3 | Discussion and limitations

Section 5.3.1's results shows the beveled tip needle model is able to fit an experimental measure of final position. More detailed studies need to be carried out in order to validate the bevel needle model in different target positions and different phantom and needle parameters. For the moment, there's no guarantee this model can be used to fit the experimental data of an entire needle path inside a phantom.

The preliminary results on the integration of the bevel needle model into the inverse simulation control framework presents interesting insights. A first observation is that the inverse simulation control framework was able to converge a similar final needle position error with symmetric or beveled tip needles. So the finite differences method described in section 3.3 is able to handle a bevel needle model without

modifications. The high frequencies observed for the bevel needle results indicates, however, the beveled tip needle usage did not improve the needle insertion procedure as expected.

Once again, the final precision metrics differ a lot from the ones reported by Baksic et al. in their article. It would be very interesting to repeat these experiments as soon as the original precision of the inverse simulation control framework is recovered.

In order to better take advantage of the beveled tip needle, an adaptation of the control framework described in section 3.3 is possible. Instead of performing one single DoF perturbation per inverse simulation loop, it would be interesting to consider sequences of rotation and translation over each of the simulation DoFs. So a pair of needle orientation change and translation over one of the simulation axis, a needle deflection could be observed. This would provide a more meaningful influence of the beveled tip needle in the *objective constraint function* than a pure orientation perturbation. This alternative could, however increase the number of inverse simulation loops needed to compute the Jacobian of the Simulation $\mathbf{J}_6$.

It is possible as well that future results which integrate the 7 DoF Robot control framework proposed in appendix A will be able to better handle the bevel orientation. The method proposes to construct the Jacobian of the simulation in the robot joint-space $\mathbf{J}_7$ instead of it's operational space $\mathbf{J}_6$. Since a simple orientation change of the needle would still not account much for the *objective constraint function*, combinations of the robot DoF perturbations would also need to be considered.

Another interesting perspective for future works is to include an additional *objective constraint function* to account for the needle orientation along its axis. This solution is already used in the literature [43] and could enhance the control framework performance when using asymmetric needles. This could be combined to both the operational-space or the joint-space Jacobian of simulation strategies.

6 | Conclusion

The precision of the needle position needs to be as accurate as possible in percutaneous interventions. This study was developed based on Baksic et al. inverse simulation – a state-of-the-art technique which combines fast biomechanical FEM simulations and a robotic control framework to perform needle steering. Throughout this work, two improvements to this autonomous control framework were proposed. A first contribution is the integration of a predictive model of respiratory motion into the inverse simulation. As a result, an improvement in the final needle positioning was observed and respiratory disturbances were reduced. A second contribution was the integration of a beveled tip needle FEM into the inverse simulation. By itself, the usage of an asymmetric needle in the control strategy did not improve the procedure results. However, interesting perspectives exist to adapt the framework to take advantage of resulting needle deflection in the control strategy.

Both the strategies proposed rely in the state-of-the-art relevant for this study described in chapter 2. The respiratory prediction filters explored by Johl et al. for robotic radiotherapy were applied to a needle steering control framework. Results of section 5.2 presents an interesting perspective that prediction models for respiratory motion can improve the final position precision and reduce final position oscillations. This is specially important for needle guided therapies which require a precise position for a long period of time, such as tumor ablations or cryotherapy. Future research is expected to consolidate these findings and inspire prediction model applications in other control strategies for needle steering.

The bevel needle model proposed by Misra et al. is a well known model which was already used in many robotic applications. Its integration to the inverse simulation framework expresses the versatility of this strategy to control robots based in biomechanical simulations. Future works should take advantage of this versatility to investigate more biomechanics-related objectives to the control of robotic manipulators. This is a differential feature of the inverse simulation proposed by Baksic and his research team .

In addition to the discussion of sections 5.2.5 and 5.3.3, its important to clear that the proposed study was still not validated in physical experiments. One of the intended outcomes of this master thesis was to advance towards moving-phantom experiments. Unfortunately, time and public health constraints delayed these plans.

Hopefully in the upcoming months the new robot model proposed in appendix A will enable tests in static and moving phantoms using a redundant collaborative robotic manipulator.

7 | Bibliography

[1] T. L. De Jong, N. J. van de Berg, L. Tas, A. Moelker, J. Dankelman, and J. J. van den Dobbelsteen, "Needle placement errors: Do we need steerable needles in interventional radiology?," *Medical Devices: Evidence and Research*, vol. 11, pp. 259–265, 2018.

[2] P. Baksic, H. Courtecuisse, C. Duriez, and B. Bayle, "Robotic needle insertion in moving soft tissues using constraint-based inverse Finite Element simulation," *ICRA 2020-IEEE International Conference on Robotics and Automation*, pp. 1–7, 2020.

[3] Y. Adagolodjo, L. Goffin, M. De Mathelin, and H. Courtecuisse, "Inverse real-time Finite Element simulation for robotic control of flexible needle insertion in deformable tissues," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-Novem, pp. 2717–2722, 2016.

[4] R. J. Webster, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura, "Nonholonomic modeling of needle steering," *International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 509–525, 2006.

[5] R. E. Kalman and R.Buey, "A new approach to linear filtering and prediction theory.," *Trans. ASME, Journal of Basic Engineering*, vol. 83, no. Series D, pp. 95–108, 1961.

[6] J. A. Engh, G. Podnar, S. Y. Khoo, and C. N. Riviere, "Flexible needle steering system for percutaneous access to deep zones of the brain," *Bioengineering, Proceedings of the Northeast Conference*, vol. 2006, pp. 103–104, 2006.

[7] J. Chevrie, "Flexible Needle Steering using Ultrasound Visual Servoing," 2017.

[8] B. Fallahi, M. Khadem, C. Rossa, R. Sloboda, N. Usmani, and M. Tavakoli, "Extended bicycle model for needle steering in soft tissue," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem, pp. 4375–4380, 2015.

[9] S. Misra, K. Reed, B. Schafer, K. Ramesh, and A. Okamura, "Mechanics of Flexible Needles Robotically Steered through Soft Tissue," *The International Journal of Robotics Research*, vol. 29, pp. 1640–1660, nov 2010.

[10] J. Chevrie, A. Krupa, and M. Babel, "Needle steering fusing direct base manipulation and tip-based control," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2016-June, pp. 4450–4455, 2016.

[11] S. Misra, K. B. Reed, B. W. Schafer, K. T. Ramesh, and A. M. Okamura, "Observations and models for needle-tissue interactions," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2687–2692, 2009.

[12] Y. Adagolodjo, L. Goffin, M. De Mathelin, and H. Courtecuisse, "Robotic Insertion of Flexible Needle in Deformable Structures Using Inverse Finite-Element Simulation," *IEEE Transactions on Robotics*, vol. 35, no. 3, pp. 697–708, 2019.

[13] S. P. DiMaio and S. E. Salcudean, "Needle Insertion Modeling and Simulation," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 5, pp. 864–875, 2003.

[14] S. P. DiMaio and S. E. Salcudean, "Interactive simulation of needle insertion models," *IEEE transactions on biomedical engineering*, vol. 52, no. 7, pp. 1167–1179, 2005.

[15] N. Chentanez, R. Alterovitz, D. Ritchie, L. Cho, K. K. Hauser, K. Goldberg, J. R. Shewchuk, and J. F. O'Brien, "Interactive simulation of surgical needle insertion and steering," *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 1–10, 2009.

[16] O. Goksel, E. Dehghan, and S. E. Salcudean, "Modeling and simulation of flexible needles," *Medical Engineering and Physics*, vol. 31, no. 9, pp. 1069–1078, 2009.

[17] C. Duriez, C. Guébert, M. Marchal, S. Cotin, and L. Grisoni, "Interactive simulation of flexible needle insertions based on constraint models," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5762 LNCS, no. PART 2, pp. 291–299, 2009.

[18] H. Courtecuisse, J. Allard, P. Kerfriden, S. P. Bordas, S. Cotin, and C. Duriez, "Real-time simulation of contact and cutting of heterogeneous soft-tissues," *Medical Image Analysis*, vol. 18, no. 2, pp. 394–410, 2014.

[19] J. Gangloff, R. Ginhoux, M. de Mathelin, L. Soler, and J. Marescaux, "Model predictive control for compensation of cyclic organ motions in teleoperated laparoscopic surgery," *IEEE Transactions on Control Systems Technology*, vol. 14, pp. 235–246, mar 2006.

[20] A. Shademan, R. S. Decker, J. D. Opfermann, S. Leonard, A. Krieger, and P. C. Kim, "Supervised autonomous robotic soft tissue surgery," *Science Translational Medicine*, vol. 8, no. 337, 2016.

[21] L. Ott, F. Nageotte, P. Zanne, and M. De Mathelin, "Robotic assistance to

flexible endoscopy by physiological-motion tracking," *IEEE Transactions on Robotics*, vol. 27, no. 2, pp. 346–359, 2011.

[22] Lingbo Cheng, "Control of Teleoperation Systems for Beating-Heart Surgery," p. 55, 2019.

[23] G. J. Vrooijink, A. Denasi, J. G. Grandjean, and S. Misra, "Model predictive control of a robotically actuated delivery sheath for beating heart compensation," *The International Journal of Robotics Research*, vol. 36, pp. 193–209, feb 2017.

[24] M. A. Clifford, F. Banovac, E. Levy, and K. Cleary, "Assessment of hepatic motion secondary to respiration for computer assisted interventions," *Computer Aided Surgery*, vol. 7, no. 5, pp. 291–299, 2002.

[25] H. Arenbeck, L. Wittschier, D. Kügler, and D. Abel, "Control methods for robot-based predictive compensation of respiratory motion," *Biomedical Signal Processing and Control*, vol. 34, pp. 16–24, 2017.

[26] H. D. Kubo and B. C. Hill, "Respiration gated radiotherapy treatment: A technical study," *Physics in Medicine and Biology*, vol. 41, no. 1, pp. 83–91, 1996.

[27] C. Cavedon, "Real-time control of respiratory motion: Beyond radiation therapy," *Physica Medica*, vol. 66, no. September, pp. 104–112, 2019.

[28] W. Kilby, M. Naylor, J. R. Dooley, C. R. Maurer, and S. Sayeh, *A Technical Overview of the CyberKnife System.* Elsevier Inc., 2020.

[29] A. Jöhl, S. Ehrbar, M. Guckenberger, S. Klöck, A. Mack, M. Meboldt, M. Zeilinger, S. Tanadini-Lang, and M. Schmid Daners, "The ideal couch tracking system—Requirements and evaluation of current systems," *Journal of Applied Clinical Medical Physics*, vol. 20, no. 10, pp. 152–159, 2019.

[30] A. Schweikard, H. Shiomi, and J. Adler, "Respiration tracking in radiosurgery," *Medical Physics*, vol. 31, no. 10, pp. 2738–2741, 2004.

[31] C. Ozhasoglu, C. B. Saw, H. Chen, S. Burton, K. Komanduri, N. J. Yue, S. M. Huq, and D. E. Heron, "Synchrony - Cyberknife Respiratory Compensation Technology," *Medical Dosimetry*, vol. 33, no. 2, pp. 117–123, 2008.

[32] J. J. Nuyttens, J. B. Prévost, J. Praag, M. Hoogeman, R. J. Van Klaveren, P. C. Levendag, and P. M. Pattynama, "Lung tumor tracking during stereotactic radiotherapy treatment with the CyberKnife: Marker placement and early results," *Acta Oncologica*, vol. 45, no. 7, pp. 961–965, 2006.

[33] R. Urschel, Harold, Kresl, John, Luketich, James, Papiez, Lech, Timmerman, Robert, Schulz, *Treating Tumors that Move with Respiration.* 2007.

[34] B. T. Collins, S. Vahdat, K. Erickson, S. P. Collins, S. Suy, X. Yu, Y. Zhang, D. Subramaniam, C. A. Reichner, I. Sarikaya, G. Esposito, S. Yousefi, C. Jamis-

Dow, F. Banovac, and E. D. Anderson, "Radical CyberKnife radiosurgery with tumor tracking: An effective treatment for inoperable small peripheral stage i non-small cell lung cancer," *Journal of Hematology and Oncology*, vol. 2, pp. 1–9, 2009.

[35] A. C. Koong, E. Christofferson, Q. T. Le, K. A. Goodman, A. Ho, T. Kuo, J. M. Ford, G. A. Fisher, R. Greco, J. Norton, and G. P. Yang, "Phase II study to assess the efficacy of conventionally fractionated radiotherapy followed by a stereotactic radiosurgery boost in patients with locally advanced pancreatic cancer," *International Journal of Radiation Oncology Biology Physics*, vol. 63, no. 2, pp. 320–323, 2005.

[36] B. Fallahi, L. Cheng, and M. Tavakoli, *State observation and feedback control in robotic systems for therapy and surgery.* 2020.

[37] C. N. Riviere, A. Thakral, I. I. Iordachita, G. Mitroi, and D. Stoianovici, "Predicting respiratory motion for active canceling during percutaneous needle insertion," *Annual Reports of the Research Reactor Institute, Kyoto University*, vol. 4, pp. 3477–3480, 2001.

[38] A. Jöhl, S. Ehrbar, M. Guckenberger, S. Klöck, M. Meboldt, M. Zeilinger, S. Tanadini[U+2010]Lang, and M. Schmid Daners, "Performance comparison of prediction filters for respiratory motion tracking in radiotherapy," *Medical Physics*, vol. 47, pp. 643–650, feb 2020.

[39] C. N. Riviere, J. Gangloff, and M. De Mathelin, "Robotic compensation of biological motion to enhance surgical accuracy," *Proceedings of the IEEE*, vol. 94, no. 9, pp. 1705–1715, 2006.

[40] R. Ginhoux, J. Gangloff, M. de Mathelin, L. Soler, M. M. Arenas Sanchez, and J. Marescaux, "Active filtering of physiological motion in robotized surgery using predictive control," *IEEE Transactions on Robotics*, vol. 21, no. 1, pp. 67–79, 2005.

[41] P. Moreira, M. Abayazid, and S. Misra, "Towards physiological motion compensation for flexible needle interventions," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem, pp. 831–836, 2015.

[42] G. J. Vrooijink, M. Abayazid, S. Patil, R. Alterovitz, and S. Misra, "Needle path planning and steering in a three-dimensional non-static environment using two-dimensional ultrasound images," *The International journal of robotics research*, vol. 33, no. 10, pp. 1361–1374, 2014.

[43] J. Chevrie, N. Shahriari, M. Babel, A. Krupa, and S. Misra, "Flexible Needle Steering in Moving Biological Tissue with Motion Compensation Using Ultrasound and Force Feedback," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2338–2345, 2018.

[44] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," *IEEE 2000 Adaptive Systems for Signal Processing, Communica-*

tions, and Control Symposium, AS-SPCC 2000, no. February 2000, pp. 153–158, 2000.

[45] J. Chevrie, A. Krupa, M. Babel, J. Chevrie, A. Krupa, M. Babel, N. Steering, F. Direct, B. Manipulation, J. Chevrie, A. Krupa, and M. Babel, "Needle Steering Fusing Direct Base Manipulation and Tip-based Control To cite this version : HAL Id : hal-01304860 Needle Steering Fusing Direct Base Manipulation and Tip-based Control," 2016.

[46] H. W. Kuhn and A. W. Tucker, "Nonlinear programming," *Traces and Emergence of Nonlinear Programming*, no. x, pp. 247–258, 2014.

[47] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*, pp. 1211–1216 vol.2, IEEE, 1991.

[48] Y. Nakamura, *Advanced robotics: redundancy and optimization*. Addison-Wesley Longman Publishing Co., Inc., 1990.

[49] D. Ruan and P. Keall, "Online prediction of respiratory motion: multidimensional processing with low-dimensional feature learning," *Physics in Medicine and Biology*, vol. 55, pp. 3011–3025, jun 2010.

[50] F. Ernst, *Compensating for quasi-periodic motion in robotic radiosurgery*, vol. 9781461419. 2012.

[51] J. Shlens, "A Tutorial on Principal Component Analysis," 2014.

[52] M. Avriel and D. J. Wilde, "Optimally proof for the symmetric fibonacci search technique," *The Fibonacci Quarterly*, vol. 4, no. 3, pp. 265–269, 1966.

[53] F. Ernst, R. Dürichen, A. Schlaefer, and A. Schweikard, "Evaluating and comparing algorithms for respiratory motion prediction," *Physics in Medicine and Biology*, vol. 58, no. 11, pp. 3911–3929, 2013.

[54] J. M. Hudson, L. Milot, C. Parry, R. Williams, and P. N. Burns, "Inter- and Intra-Operator Reliability and Repeatability of Shear Wave Elastography in the Liver: A Study in Healthy Volunteers," *Ultrasound in Medicine and Biology*, vol. 39, no. 6, pp. 950–955, 2013.

Appendices

A | Constraint based IK of a 7 DoF Robot

Another contribution of this study is a constraint-based solution to a redundant robot inverse kinematics (IK). This work was developed to integrate a new robotic manipulator of the ICube laboratory into biomechanical simulations. Even though promising tests were performed in phantom simulations, the contribution of this solution to the needle insertion of moving objects has not yet been addressed.

As mentioned in section 1.1, at the end of the inverse simulation a cartesian target position which reduce the procedure objective constraints is sent to the robotic device. This solution supposes the robot possesses an inverse kinematics solver to find adequate joint positions to achieve the target position in its operational space. For a 6 DoF robot, such as the Mitsubishi-MRV1A used by Adagolodjo, an analytical solution to the IK exist to non-singular operational space targets. However, for redundant robots with 7 DoF there is usually multiple solutions to a same cartesian target. A solution is to take account of the system's objective constraints and it's task optimization framework to find an optimal solution in the joint-space of the robot.

The 7 DoF robotic manipulator model used through this study is a Kuka IIWA 14Kg collaborative manipulator. This is one of the robotic arms the ICube laboratory possesses and it's expected to replace the MRV1A in future practical experiments. The manipulator direct kinematics and its jacobian were previously computed and validated in a recent work of the laboratory.

A.1 | Operational Space Finite Differences

In previous studies, the inverse simulation performs 12 simulation loops to derive a Jacobian matrix between cartesian displacements of the end effector $\Delta\chi$ and the objective functions ($\mathbf{e}$). A positive and a negative displacement of the end effector over each of it's translation and orientation DoFs is performed. Then, after each displacement, an specific row of the jacobian matrix can be estimated using the finite difference method between the positive and negative displacement of a same DoF.

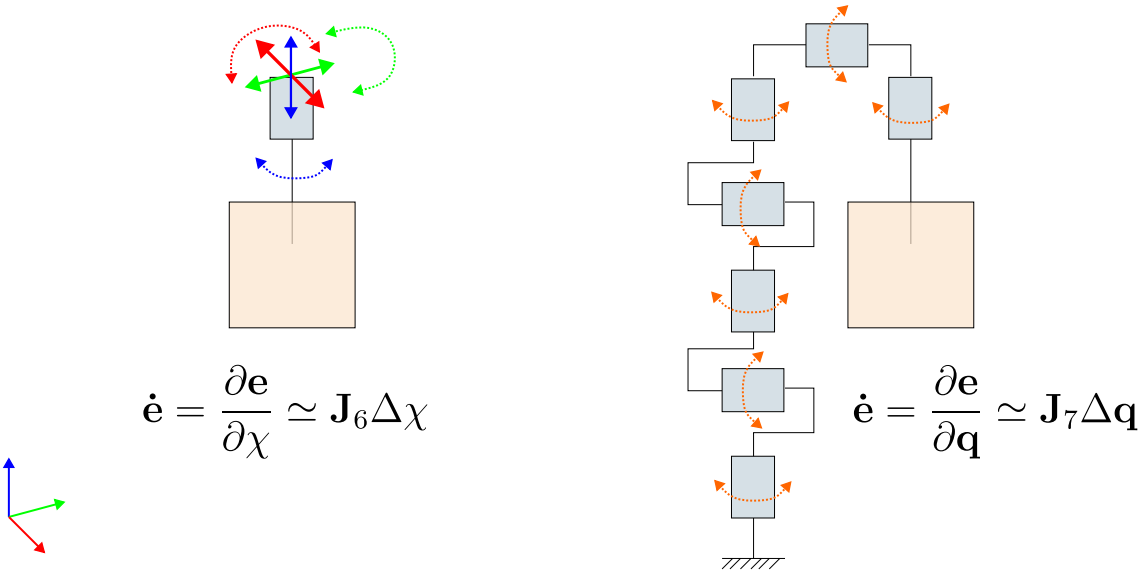

Figure A.1: Finite differences method applied to a needle insertion robot end effector and to each of the robot's joints, on the left and right sides respectively. The first derivative of the objective constraints for this configuration $\dot{\mathbf{e}}$ is related to the end effector position though the Jacobian matrix $\mathbf{J_6}$. Similarly, $\dot{\mathbf{e}}$ is related to the robot joint positions through the Jacobian matrix $\mathbf{J}_7$

The left portion of figure A.1 expresses an intuitive view of the robotic end effector displacements during a needle insertion. Equation A.1 details the computation of the objective function jacobian based in the finite differences method.

$$\mathbf{J}_6^j = \frac{E(\mathbf{q}, \chi + \Delta\chi^j) - E(\bar{\mathbf{q}}, \bar{\chi} - \Delta\chi^j)}{2||\Delta\chi^j||} \tag{A.1}$$

A.2 | Robot Joint Space Finite Differences

In order to solve this jacobian matrix in the joint-space of the robotic manipulator, an analogous finite-differences approach can be used. At each inverse simulation loops, a perturbation of one of the robot axis will be performed (Δq^j) and its influence with respect to the objective functions will be observed $\mathbf{e}$. So with 14 inverse simulation loops, a positive and a negative disturbance of each axis will be accounted for and their jacobian with respect to the objective constraints can be computed. Then the Jacobian matrix between the objective constraints and the redundant robot can be computed by finite differences as expressed in equation A.2.

$$\mathbf{J}_7^j = \frac{E(\mathbf{q} + \Delta q^j, \chi) - E(\bar{\mathbf{q}} - \Delta q^j, \bar{\chi})}{2||\Delta q^j||} \tag{A.2}$$

Then, the inversion of the Jacobian matrix can also be performed using a Tikhonov regularization, as expressed in equations 3.7 and 3.8. We denote once again the regularization coefficient as r. The solution of the regularized pseudoinverse is $\mathbf{J}_{7\alpha}^{+}$. Finally, optimal robotic joint targets $\mathbf{q}^{(i+1)}$ for the future time step $i+1$ can be computed with respect to the objective constraint functions and a control gain $\mathbf{k}$. The control gain is applied over the objective functions using a Hadamard product $\odot$.

$$\mathbf{q}^{(i+1)} = \mathbf{q}^{(i)} - \mathbf{J}_{7r}^{+}(\mathbf{k} \odot \mathbf{e}^{(i)}) \tag{A.3}$$

A.3 | Integration to the Inverse Simulation

The addition of a 7 DoF robot manipulator model can provide advantages to the inverse simulation control framework. First, the finite differences of *objective constraint functions* in joint space expressed in equation A.2 may provide different end-effector positions. This is specially important for asymmetric needle insertions, in which the needle orientation along it's axis is relevant for the needle insertion.

While a 6 DoF robot may be able to adapt its effector orientation to steer the beveled tip needle, it might not be able to reach its target position without violation of the entry *point objective constraint function* $\mathbf{e}_e$. A 7 DoF robot will dispose of more joint-space alternatives to reach a same operational space target with specific needle orientation along it's axis.